Datenbankgestützte Substanzbeschaffung in der forschenden Chemieindustrie – Ein algorithmischer Optimierungsansatz

Von der Mercator School of Management

– Fakultät für Betriebswirtschaftslehre –

der Universität Duisburg-Essen

zur Erlangung des akademischen Grades

eines Doktors der Wirtschaftswissenschaften (Dr. rer. oec.)

genehmigte Dissertation

von

Dipl.-Oec. Ernst-Georg Schmid

aus

Duisburg

Referent:	Prof. Dr. Peter Chamoni
Korreferent:	Prof. Dr. Alf Kimms
Tag der mündlichen Prüfung:	21. Juli 2010

Danksagung

Die vorliegende Arbeit entstand im Zeitraum von 2007 bis 2010 am Lehrstuhl für Wirtschaftsinformatik und Operations Research der Mercator School of Management an der Universität Duisburg-Essen und mit Unterstützung der Abteilung Science & Technology der Bayer Business Services GmbH.

Prof. Dr. Peter Chamoni danke ich besonders für das Wagnis, mich eine Dekade nach meiner Diplomprüfung als externen Doktoranden anzunehmen.

Prof. Dr. Alf Kimms danke ich für die Übernahme des Zweitgutachtens.

Bei der Bayer Business Services GmbH gilt mein Dank Dr. Michael Schimeczek und Dr. Susanne Koppelmann für ihre tatkräftige Unterstützung.

Weiterhin danke ich Prof. Dr. Achim Zielesny vom Fachbereich Angewandte Naturwissenschaften der Fachhochschule Gelsenkirchen und Prof. Dr. Norbert Haider vom Institut für Pharmazeutische Chemie der Universität Wien dafür, dass sie jahrelang geduldig meine Fragen ertragen (und beantwortet) haben.

For all the insightful discussions about the handling of chemical information in database systems in general and the technical support for programming with OpenBabel I would like to thank Rajarshi Guha, Peter Murray Rust, Jérôme Pansanel, Chris Morley, Craig A. James, Rich Apocada, Andrew Dalke, and all the other members of the Blue Obelisk Group that I forgot to mention here.

Weiterhin möchte ich mich bei Dr. Oliver Herd von der Chemcollect GmbH, Hans Kraut von der Infochem GmbH und Alan Whittle von Maybridge für die Überlassung von Testdaten bedanken.

Kurzfassung

Die Speicherung und Suche chemischer graphischer Datentypen wie Strukturen und Reaktionen in relationalen Datenbanksystemen ist ein in Wissenschaft und Industrie etabliertes Verfahren. Aufgrund der rechenintensiven Algorithmen zur Erkennung von (Sub)Graphen-Isomorphismus benutzen solche Systeme in der Regel schnellere Selektionsmechanismen, um die Menge potentieller Kandidaten bereits im Vorfeld einzuschränken.

Dabei werden verbreitet Selektionsmechanismen eingesetzt, die auf numerischen und binären Vektoren, Fingerprints genannt, basieren, mit einer klaren Dominanz binärer Fingerprints aufgrund ihrer Geschwindigkeitsvorteile bei bitweisen Operationen und der besseren Speichereffizienz. Die beiden am Häufigsten eingesetzten binären Fingerprints sind einerseits Pfad-generiert, andererseits Wörterbuch-generiert, wobei beide spezifische Schwächen, insbesondere *blinde Stellen*, aufweisen.

Um diese Schwächen zu überwinden, benutzt die Pgchem::Tigress Erweiterung für das objektrelationale Datenbankmanagementsystem PostgreSQL einen hybriden binären Fingerprint, der aus einem invarianten Pfad-generierten Teil und einem Substruktur-generierten Teil besteht, welcher extern durch ein Wörterbuch von Substrukturmustern konfiguriert werden kann.

Diese Arbeit stellt einen neuartigen Ansatz vor, um für beliebige Strukturdaten mittels dynamischer diskreter Optimierung die optimierte Konfiguration des Wörterbuchs für den Substruktur-generierten Teils des Fingerprints zu finden.

Mittels des Einsatzes des in dieser Arbeit entwickelten Verfahrens kann die notwendige Rechenleistung zum Betrieb eines chemischen Informationssystems um durchschnittlich 42 Prozent reduziert werden. Durch den so verbesserten Anfragedurchsatz lassen sich der Umstieg auf die nächsthöhere verfügbare Leistungsstufe eines Servers vermeiden und so signifikante Opportunitätserlöse bei den Betriebskosten realisieren.

Abstract

The storage and retrieval of chemical graphical datatypes such as structures and reactions in relational database systems is a common technique used in academia and industry alike. Due to the computationally intensive algorithms used for (sub)graph-isomorphism detection, such systems commonly use faster screening mechanisms in order to reduce the set of potentional match positives before applying aforementioned algorithms.

Widely used screening mechanisms are based on numerical and binary vectors, called fingerprints, with a clear dominance of binary fingerprints due to the raw speed advantage of bitwise operations and compactness in storage. The two most commonly used types of binary fingerprints are path-generated and substructure-generated, both of which have specific shortcomings, especially *blind spots*.

To overcome this shortcomings, the PGCHEM::TIGRESS chemistry extension to the POSTGRESQL object-relational database management system uses a hybrid binary fingerprint, consisting of an invariant path-generated part and an substructure-generated part which is externally configurable through a dictionary of substructure patterns.

This thesis presents a novel approach of using dynamic discrete optimization to find an optimized dictionary configuration for the substructure-generated part of the fingerprint for arbitrary sets of structural data.

By means of applying the method developed in this thesis, the computational power neccessary to run a chemical information system can be reduced by 42 percent on average. By improving the query throughput, upgrading the server hardware to the next level of computational power can be avoided and thus opportunity revenues of the operating costs are realized.

Abkürzungsverzeichnis

ACD	Available Chemicals Directory
ACS	American Chemical Society
ANOVA	Analysis of Variance
API	Application Programming Interface
ArbnErfG	Gesetz über Arbeitnehmererfindungen
ASP	Application Service Provider
B2B	Business-to-Business
BBS	Bayer Business Services GmbH
BtMG	Betäubungsmittelgesetz
CAS	Chemical Abstracts Service
ChEBI	Chemical Entities of Biological Interest
ChemACX	Available Chemicals Exchange
CRO	Clinical Research Organisation
DOE	Design of Experiments
DoS	Denial Of Service
EINECS	European INventory of Existing Commercial Chemical Substances
ELINCS	European LIst of Notified Chemical Substances
ELN	Electronic Lab Notebook
FuE	Forschung und Entwicklung

GA	Genetischer Algorithmus
GCP	Good Clinical Practice
GHS	Globally Harmonized System of Classification and Labelling of Chemicals
GIN	Generalized Inverted Index
GiST	Generalized Search Tree
GLP	Good Laboratory Practice
GMP	Good Manufacturing Practice
InChI	IUPAC International Chemical Identifier
IRM	Information Resource Management
IT	Informationstechnologie
IUPAC	International Union of Pure and Applied Chemistry
JDK	Java™ Development Kit
LP	Lineares Programm
MBB	Minimum Bounding Box
MBR	Minimum Bounding Rectangle
NP	nichtdeterministisch polynomiale Zeit
OFAT	One-Factor-at-a-Time
OR	Operations Research
PAK	Polyzyklische aromatische Kohlenwasserstoffe
PatG	Patentgesetz
PhRMA	Pharmaceutical Research and Manufacturers of America
RDBMS	Relational Database Management System

ROI	Return on Investment
SaaS	Software as a Service
SAR	Set of All Rings
SIMD	Single Instruction Multiple Data
SMARTS	SMILES Arbitrary Target Specification
SMILES	Simplified Molecular Input Line Entry System
SQL	Structured Query Language
SSSR	Smallest Set of Smallest Rings
VC	Verfügbare Chemikalien
VCI	Verband der chemischen Industrie
XML	EXtensible Markup Language

Symbolverzeichnis

C_T	Tanimoto Index
D_S	Soergel Distanz
D_q	Anfragedurchsatz
$E(h)$	Erwartungswert
F	Zielfunktion
F'	Ersatzzielfunktion
N	Größe der Grundgesamtheit
P	Anteil der Merkmalsträger in der Grundgesamtheit (Stichprobenanteil)
$P(x)$	Wahrscheinlichkeitsfunktion
Q	Anteil der Nichtmerkmalsträger in der Grundgesamtheit
S	Selektivität
T_B	Basistabelle der Optimierung
T_Q	Quarantänetabelle
Δ_S	Selektivitätsveränderung
Γ	Dekodierfunktion
χ^2	Chi-Quadrat-Wert
$\mathbb{N}$	Menge der natürlichen Zahlen ohne Null
$\mathbb{N}_0$	Menge der natürlichen Zahlen mit Null
$\overline{x}$	Arithmetisches Mittel
ϕ	Fitnessfunktion
d	Breite des Konfidenzintervalls
n_0	Stichprobengröße ohne Endlichkeitskorrektur
n_{korr}	Stichprobengröße mit Endlichkeitskorrektur
r^2	Determinationskoeffizient

s	Standardabweichung
t	Wert der Standardnormalverteilung für das Signifikanzniveau α
t_{INSERT}	Laufzeit für INSERT
t_{Run}	Laufzeit eines Testsuchenexperiments

Tabellenverzeichnis

Abbildungsverzeichnis

Inhaltsverzeichnis

Einleitung, Ziel und Aufbau der Arbeit

Eine Studie der Unternehmensberatung Bain & Company über den Return on Investment (ROI) neu entwickelter Medikamente kommt zu dem Ergebnis: „When the costs of failed prospective drugs are factored in, the actual cost for discovering, developing and launching a single new drug has risen to nearly $1.7 billion. That's a 55 % increase over the average commercialization cost for the five years from 1995 to 2000“ [BC03] und macht sinkende Produktivität bei gleichzeitig steigenden Kosten der notwendigen Forschung und Entwicklung (FuE) dafür verantwortlich.

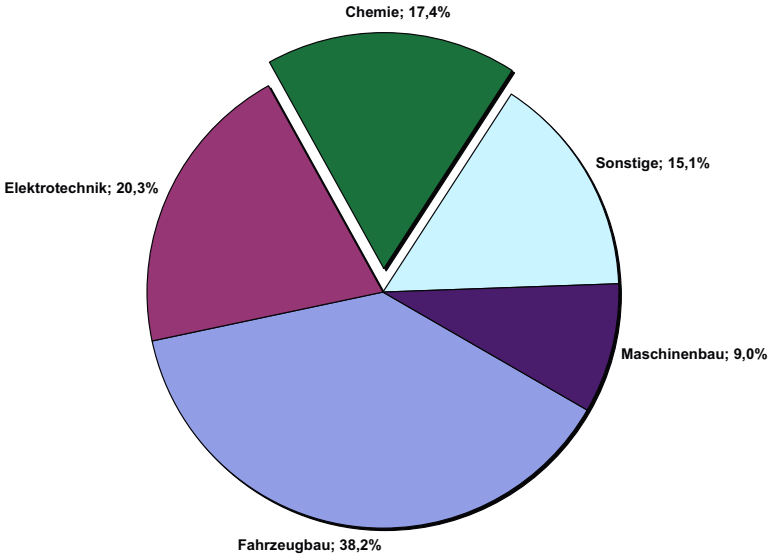


Abbildung 1.1: FuE-Aufwendungen nach Branchen für die Bundesrepublik Deutschland im Jahr 2006
Quelle: Eigene Darstellung auf Basis des FuE-Datenreport 2008 [Sti08, S. 14]

Die Pharmaceutical Research and Manufacturers of America (PhRMA) geben den jährlichen Zuwachs ihrer FuE-Ausgaben seit 1970 mit 12,3 Prozent an: „R&D spending has been growing at an average compounded rate of 12.3 % since 1970“. [Mun09, S. 962]

Unter der Überschrift „Trotz steigender FuE-Aufwendungen in der Chemie: FuE-Personaleinsatz rückläufig“ stellt die Wissenschaftsstatistik gGmbH fest: „Die Zunahme des Wissenschaftleranteils am FuE-Personal und der rückläufige Anteil der Personalaufwendungen an den internen FuE-Aufwendungen hat dazu geführt, dass von 1995 bis 2003 der FuE-Einsatz je FuE-Beschäftigtem in der Chemie (Pharmazie) von 98,8 Tsd. Euro (97,6 Tsd. Euro) inzwischen auf 162,8 Tsd. Euro (201,5 Tsd. Euro) gestiegen ist. . .“ [FuE07].

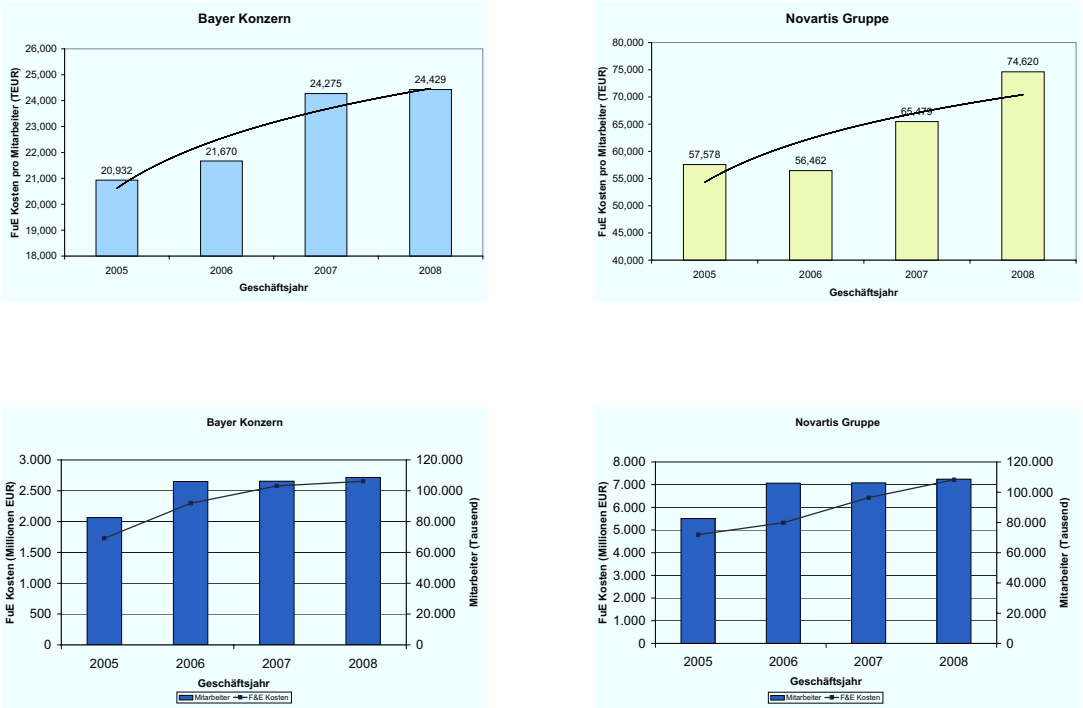


Abbildung 1.2: FuE-Aufwendungen des Bayer Konzerns und der Novartis Gruppe pro Mitarbeiter und absolut von 2005 bis 2008
Quelle: Eigene Darstellung auf Basis der Geschäftsberichte des Bayer Konzerns für die Jahre 2006 bis 2008 [Bay06, Bay07, Bay08] und des Geschäftsberichts der Novartis Gruppe 2008 [Nov08]

Dieser Trend steigender FuE-Aufwendungen pro Mitarbeiter spiegelt sich exemplarisch auch in den FuE-Aufwendungen pro Mitarbeiter des Bayer Konzerns und der Novartis Gruppe für die Jahre 2006 bis 2008 wider, wie in Abbildung 1.2 graphisch dargestellt.

Die ebenfalls in Abbildung 1.2 dargestellte absolute Entwicklung von FuE-Aufwendungen und Mitarbeiterzahl der genannten Unternehmen in diesem Zeitraum zeigt, dass der relative Anstieg der FuE-Aufwendungen pro Mitarbeiter nicht durch Personalreduzierung erklärbar ist.

Tabelle 1.1: Kennzahlen zum FuE-Personal in Unternehmen der Chemie von 1995 bis 2003
Quelle: [FuE07, Tab. 4]

Wirtschaftssektor / Jahr	Beschäftigte in FuE (Vollzeitäquivalent)		interne FuE-Aufw. je Vollzeitäquivalent	Personalaufwand je Vollzeitäquivalent
	insgesamt	Anteil Wissenschaftler u. Ingenieure		
	Anzahl	%	Tsd. €	
	1	2	3	4
Insgesamt				
2003	294 377	54,3	128,2	75,1
Chemische Industrie				
1995	49 012	26,1	98,8	58,6
1997	47 241	26,5	114,9	64,9
1999	44 103	27,9	129,4	68,3
2001	42 001	29,8	140,9	73,3
2003	41 976	32,0	151,2	76,6
H. v. pharmazeutischen Erzeugnissen				
1995	12 804	32,3	97,6	58,2
1997	17 007	30,7	111,3	63,0
1999	15 232	35,0	137,2	68,8
2001	15 512	37,6	146,8	70,3
2003	16 904	34,9	180,8	78,5

Quelle: Stifterverband Wissenschaftsstatistik

Während also die Aufwendungen für FuE der chemischen Industrie, die im Jahr 2006, wie aus Abbildung 1.1 ersichtlich, einen Anteil von 17,4 Prozent der gesamten FuE-Aufwendungen in der Bundesrepublik Deutschland bestritt, insgesamt steigen, wird gleichzeitig das FuE-Personal reduziert.

Tabelle 1.1 zeigt die Reduktion der Beschäftigten in FuE von 1995 bis 2003 und eine parallele Verschiebung der Personalstruktur hin zu höher qualifizierten Mitarbeitern: „. . .in dieser Zeit [1995 bis 2003; Anm. d. Verf.] ist auch die Qualifikation des FuE-Personals

gestiegen, denn der Anteil der Wissenschaftler und Ingenieure hat zugenommen: in der Chemie von 26 % auf 32 %, in der Pharmazie von 32 % auf 35 %.“ [FuE07].

Zur Kompensation dieses Trends steigender Aufwendungen wird der Automatisierungsgrad in der chemischen Forschung erhöht, um diese höher qualifizierten Mitarbeiter von Routinetätigkeiten zu entlasten und sie mit Werkzeugen zur effizienten Unterstützung des Forschungsprozesses zu versehen: „Dies lässt darauf schließen, dass in der chemischen Industrie gegenüber dem gesamten Wirtschaftssektor kapitalintensiver geforscht wird...“ [FuE07].

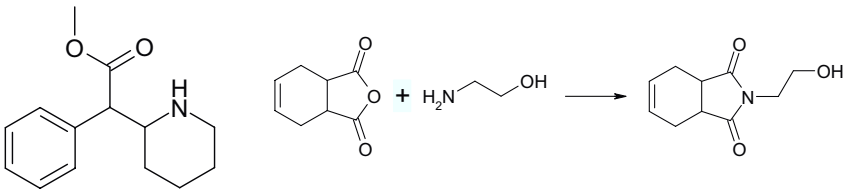


Abbildung 1.3: Beispiele chemischer, graphischer Datentypen: Struktur und Reaktion
Quelle: Eigene Darstellung

Ein wesentlicher Aspekt solcher den Forschungsprozess unterstützender Werkzeuge ist die Fähigkeit chemische, graphische Datentypen wie Strukturen und Reaktionen, wie in Abbildung 1.3 exemplarisch gezeigt, in relationalen Datenbanksystemen (Relational Database Management System (RDBMS)) verwalten zu können.

Die Verwaltung chemischer, graphischer Datentypen in RDBMS ist ein in Wissenschaft und Industrie etabliertes Verfahren: „One of the most fundamental tasks of chemoinformatics is the rapid search of large repositories of molecules containing millions of compounds...“ [SB07, S. 952].

Handelsübliche RDBMS können nativ allerdings nicht mit solchen Datentypen umgehen und werden daher mit Zusatzmodulen um diese Fähigkeit erweitert. Abbildung 1.4 zeigt schematisch den Aufbau einer solchen Erweiterung.

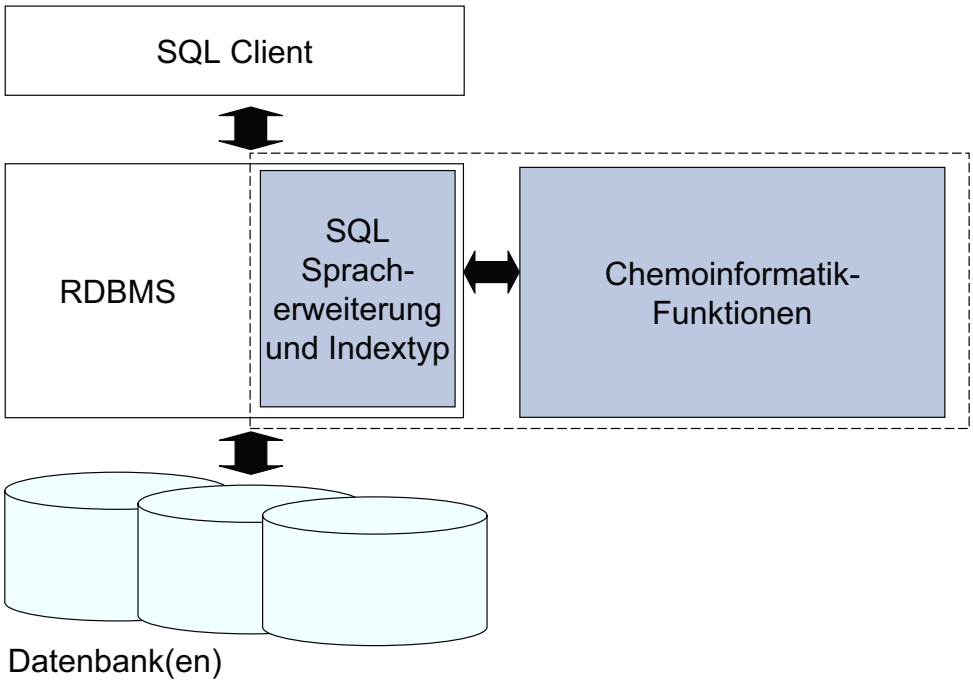


Abbildung 1.4: Prinzipieller Aufbau einer Datenbankcartridge
Quelle: Eigene Darstellung

Während vorgenannte Zusatzmodule seit den 1980er Jahren kommerziell erhältlich sind, gibt es erst seit den 2000er Jahren auch Alternativen in Form freier Software. All diesen Systemen ist gemein, dass sie Mechanismen implementieren, um die Laufzeitkomplexität der Überprüfung zweier ungerichteter Graphen auf (Sub)Graphen-Isomorphismus (siehe Unterabschnitt III.1.2) zu relaxieren.

Der am häufigsten eingesetzte Mechanismus ist dabei die Vorfilterung der potentiellen Treffermenge mittels binärer Deskriptorenvektoren: „To facilitate this task, one of the most practical and widely used computer representation for molecules is the binary fingerprint or binary feature vector representation [...] It is these fingerprints [...] that are used for efficiently searching large repositories.“ [SB07, S. 952].

Allerdings weisen alle diese Mechanismen individuelle Schwächen auf, die aus der Schwierigkeit resultieren, die Algorithmen zur Erzeugung solcher Deskriptorenvektoren a priori für *alle möglichen* Eingabedaten korrekt zu parametrieren.

Obwohl technisch möglich, existiert bisher noch kein Verfahren, um die freien Parameter der Erzeugung solcher Deskriptorenvektoren dynamisch an den konkreten Inhalt der Datenbank anzupassen und so *individuell* zu optimieren.

Die Konzeption, Beschreibung und Implementierung eines solchen Verfahrens sind daher Gegenstand dieser Arbeit.

Ziel der Arbeit

Das Ziel dieser Arbeit ist es, die Aufwendungen für FuE der chemischen Industrie zu reduzieren.

Direkt durch Reduktion der Kosten für den Betrieb chemischer Informationssysteme, insbesondere Katalog- und Bestellsysteme, als Teil der gesamten FuE Aufwendungen.

Indirekt durch Beschleunigung des datenbankgestützten Substanzbeschaffungsprozesses der Forschungsbereiche eines Chemieunternehmens mittels verbesserter Antwortzeiten der dort verwendeten Informationssysteme.

Teilziele

Zur Erreichung dieses Ziels werden für diese Arbeit folgende Teilziele definiert:

Beschreibung der Anwendungsgebiete und Anforderungen: Es werden die Anwendungsgebiete und Anforderungen an datenbankgestützte Informationssysteme in der forschenden chemischen Industrie vorgestellt

Beschreibung der Theorie der Verwaltung chemischer, graphischer Datentypen: Es wird die notwendige Theorie für das Verständnis der Verwaltung chemischer graphischer Datentypen in RDBMS beschrieben

Beschreibung der technischen Probleme und ihrer Ursachen: Es werden die technischen Probleme solcher auf graphischen Datentypen basierender Informationssysteme, insbesondere die suboptimale Selektivität der in der Screeningphase verwendeten Deskriptorenvektoren, mit ihren Ursachen beschrieben

Formale Beschreibung des zu lösenden Problems: Es wird eine formale Beschreibung des Problems der dynamischen kombinatorischen Optimierung von binären Deskriptorenvektoren entwickelt

Konzeption eines Verfahrens zur dynamischen kombinatorischen Optimierung: Es wird ein Verfahren konzipiert, welches die freien Parameter der Erzeugung binärer Deskriptorenvektoren dynamisch an den konkreten Inhalt der Datenbank anpasst und so für einen gegebenen Datenbestand individuell optimiert

Nachweis der Realisierbarkeit dieses Verfahrens: Es wird durch Implementierung einer Referenzsoftware nachgewiesen, dass das in dieser Arbeit beschriebene Verfahren realisierbar ist; seine Effektivität und Effizienz werden durch experimentelle Messungen überprüft

Aufbau der Arbeit

Kapitel Einleitung leitet diese Arbeit ein

Kapitel II stellt die Anwendungsgebiete datenbankgestützter Informationssysteme und die Anforderungen an das Informationsmanagement in der forschenden chemischen Industrie vor

Kapitel III enthält einen kurzen Überblick über die Theorie der Verwaltung chemischer, graphischer Datentypen in RDBMS; weiterhin werden die technischen Probleme und ihre Ursachen vorgestellt und daraus resultierend die formale Problemstellung dieser Arbeit entwickelt; es folgt die Vorstellung möglicher Lösungsansätze mit den Methoden des Operations Research (OR)

Kapitel IV beschreibt die Experimentalumgebung; es werden die für die Planung und Durchführung der Experimente verwendeten Methoden erklärt sowie die Ergebnisse der Experimente vorgestellt

Kapitel Diskussion, Zusammenfassung und Ausblick enthält eine Diskussion der Ergebnisse dieser Arbeit, fasst diese nochmals zusammen und gibt einen Ausblick auf mögliche weiterführende Arbeiten

II Spezielle Anforderungen an das Informationsmanagement in der forschenden chemischen Industrie

Aber nur suchen, ohne etwas finden zu können ist Religion. Im Umgang mit Maschinen ist das ein Problem. - *P. Glaser*

II.1 Der Beschaffungsprozess der Forschungsbereiche eines Chemieunternehmens

Ein Unternehmen der chemischen Industrie benötigt für die Herstellung seiner Produkte chemische Substanzen. Für die Produktionsbereiche handelt es sich hierbei normalerweise um Grundstoffe, aus denen dann komplexere chemische Strukturen synthetisiert werden, wobei diese Grundstoffe entweder selbst hergestellt oder bei externen Lieferanten in großen (daher auch als *bulk chemicals* bezeichnet), mittel- und langfristig planbaren Mengen eingekauft werden. Dabei ist die Anzahl der potentiell benötigten Substanzen relativ klein. Genaue Zahlen liegen nicht vor, aber ein grobes Schätzintervall reicht von den 30 000 vom Verband der chemischen Industrie (VCI) gemeldeten [Ahr01] bis zu den etwa 100 000 Substanzen in European INventory of Existing Commercial Chemical Substances (EINECS) [EIN02] und European LIst of Notified Chemical Substances (ELINCS) [ELI09]. Diese Substanzen können daher in der Regel über bekannte Bestellnummern oder normierte Produktbezeichnungen bestellt werden.

Der Beschaffungsprozess der Forschungsbereiche eines Chemieunternehmens hingegen ist fundamental anders. Dies resultiert aus der Tatsache, dass Forschung im Sinne der Exploration des Unbekannten nicht bzw. nur sehr kurzfristig planbar ist sowie aus den verwendeten Forschungsmethoden im modernen Prozess zur Entdeckung neuer Wirkstoffe (Drug Discovery Process):

1. Zielidentifikation (Wie kann man gezielt in ein biologisches System eingreifen, um eine gewünschte Wirkung zu erzielen?)
2. Zielvalidierung (Wirkt sich dieser Eingriff tatsächlich wie erwartet aus?)
3. Potentielle Wirkstoffe (*hits*) finden (Welche möglichen Wirkstoffe greifen im gewünschten Sinne ein?)
4. Führungsstruktur (*lead*) selektieren und chemisch optimieren (Welcher Wirkstoff ist der beste?)

Insbesondere bei Schritt 3 werden Fachwissen, Erfahrung und Intuition des Forschers zunehmend durch mechanisierte Verfahren unterstützt: „Today, most pharmaceutical companies use HTS as the primary engine driving lead discovery.“ [HP00, S. 445] Solche Verfahren können bis zu 300 000 Synthesen (High-Throughput Synthesis) oder Analysen (High-Throughput Screening) parallel durchführen. Abbildung II.1 zeigt exemplarisch einen HTS Laborroboter.

Abbildung II.1: Digilab® Hummingbird Plus HTS Workstation
Quelle: Digilab Inc.

In Schritt 4 müssen dann verschiedene Versionen der in Schritt 3 identifizierten potentiellen Wirkstoffe synthetisiert und getestet werden, um letztendlich die Führungsstruktur

zu selektieren und chemisch auf Wirksamkeit und Verträglichkeit hin zu optimieren: „Combined with the results from high throughput screens [Originale Fußnote entfernt; Anm. d. Verf.] and in-house libraries, this can mean having to select tens or hundreds of compounds from a collection of millions.“ [Guh05, S. 2]

Bei etwa 2×10^6 bekannten [Wil99, S. 1294] und konservativ geschätzt 10^{60} theoretisch möglichen stabilen organischen Strukturen [BMG96, S. 43] bedingen diese Vorgehensweisen, dass ein Forschungslabor oft kleine Mengen (< 1 kg) Edukte, Reagenzien, Katalysatoren et cetera kurzfristig bei Speziallieferanten bestellen muss. Solche Substanzen sind außerdem vergleichsweise teuer, so liegt zum Beispiel die Preisspanne eines Anbieters für Reaktionszwischenprodukte für die Synthese zwischen EUR 75 und EUR 650 für 1 g Substanz (vgl. [Che09b]).

McGregor und Pallai entwickeln ihren Artikel „Clustering of Large Databases of Compounds: Using the MDL "Keys" as Structural Descriptors“ [Mal97] unter der Prämisse, dass die Suche und Beschaffung geeigneter Substanzen sogar einen wesentlichen Anteil der Kosten eines Drug Discovery Projektes verursacht: „However, the cost of acquiring these can be a large part of the overall research cost for a drug-discovery project. Therefore it is worthwhile to apply the computational resources routinely used in the structure-based approaches to analyze these libraries and make a rational choice about which compounds to purchase from which sources, so as to maximize cost efficiency and number of hits generated.“ [Mal97, S. 443]

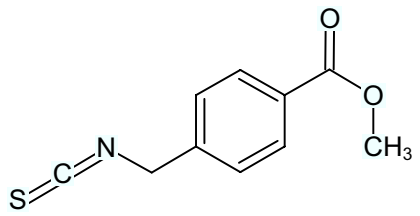
Die Bestellung ist in der Regel nicht über eine bekannte, stabile Bestellnummer möglich, und es gibt zur Zeit auch keine andere eindeutige textuelle oder numerische Kennzeichnung. Die Chemical Abstracts Service (CAS) Nummern sind aufgrund der additiven Vergabestrategie und der restriktiven Lizenzpolitik des CAS relativ oft falsch, fehlend oder mehrfach vergeben. International Union of Pure and Applied Chemistry (IUPAC)-Namen sind kompliziert zu erstellen und nicht unbedingt einmalig:„Neither a trivial name nor the systematic nomenclature, which both represent the structure as an alphanumerical (text) string, is ideal for computer processing. The reason is that various valid compound names can describe one chemical structure. . .“ [GE03, S.22]. Die IUPAC strebt dieses im zugrunde liegenden Regelwerk, dem so genannten *Blue Book*, auch gar nicht an: „The application of the general principles discussed in Section R-4 will not necessarily lead to a unique name but the name obtained should be unambiguous.“ [PPR93, R-4.0 Introduction]

Außerdem gibt es noch lokalsprachliche Eigenheiten bei den Elementnamen: „Ferner herrschen bei den Elementnamen die nationalen Gewohnheiten vor, und selbst die IUPAC-Elementwurzeln entsprechen nicht durchgängig dem für die Formelkürzel maßgebenden Namen (Beispiel Hg = Hydrargyrum, dt. Quecksilber, IUPAC-Wurzel mercur wie engl. mercury und lat. Mercurius).“ [Wik08]

Die Simplified Molecular Input Line Entry System (SMILES) Notation ist ebenfalls nicht eindeutig, da die Firma Daylight in [Wei88, WWW89] nicht die vollständige Spezifikation offengelegt hat, so dass verschiedene Implementierungen der fehlenden Teile parallel existieren. Der IUPAC International Chemical Identifier (InChI) ist relativ neu und bisher nur für Exaktsuchen geeignet. Letztlich ist auch die Summenformel (Hill-Formel) unbrauchbar, denn sie enthält keine Konformationsinformationen.

Eine exemplarische Übersicht der verschiedenen Möglichkeiten der Darstellung einer chemischen Substanz bietet Tabelle II.1.

Tabelle II.1: Verschiedene Darstellungsarten einer chemischen Substanz
Quelle: Eigene Darstellung

Konformationsformel	
	
Textdarstellung	**Beispiel**
Hill-Formel	$C_{10}H_9NO_2S$
SMILES	c1cc(C(OC)=O)ccc1CN=C=S oder auch S=C=NCc1ccc(cc1)C(=O)OC
InChI	1/C10H9NO2S/c1-13-10(12)9-4-2-8(3-5-9)6-11-7-14/h2-5H,6H2,1H3
IUPAC Name (DE)	Methyl-4-(isothiocyanatomethyl)benzolcarboxylat
IUPAC Name (EN)	Methyl 4-(isothiocyanatomethyl)benzoate oder auch 4-Isothiocyanatomethyl-benzoicacidmethylester

Aus den bereits genannten Gründen hat sich bisher allein die dort aufgeführte Konformationsformel als zuverlässiger Strukturidentifizierer in chemischen Informationssystemen etabliert, zumal diese eine jedem Chemiker geläufige Darstellungsform ist.

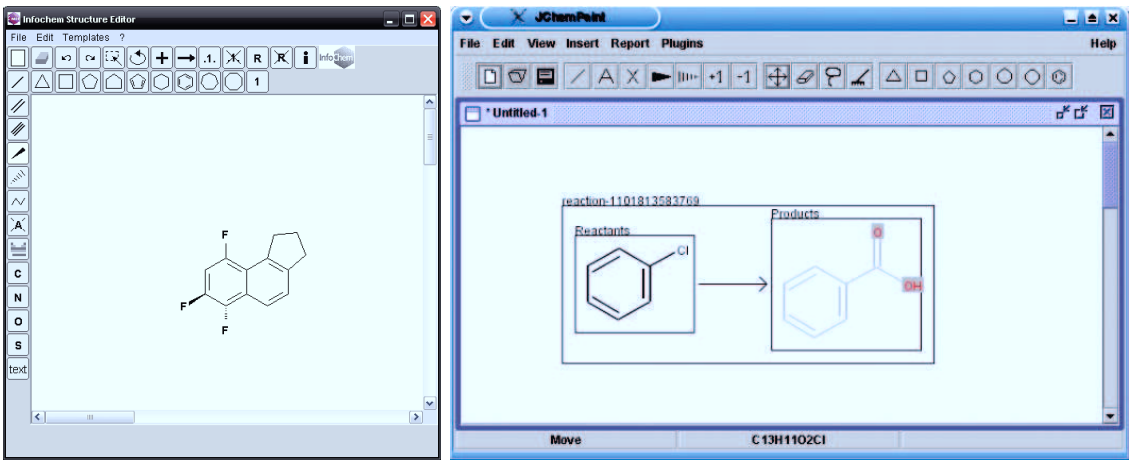


Abbildung II.2: ICEdit und JChemPaint, graphische 2D Strukturediteren
Quelle: Eigene Darstellung und [JCh09]

Zum Zeichnen solcher Konformationsformeln existiert eine Reihe graphischer Editoren, zum Beispiel die in Abbildung II.2 gezeigten ICEdit [Inf09] und JChemPaint [SKW00].

II.2 Den Forschungsprozess unterstützende Informationssysteme

II.2.1 Katalog- und Bestellsysteme

Die Bereitstellung der Katalogdaten seitens der Anbieter erfolgt neben der gedruckten Form wie exemplarisch in Abbildung II.3 gezeigt zunehmend in Form von Datenbanken. Hier kann zwischen Systemen unterschieden werden, die *in-house*, also nur im Einflussbereich des Kunden betrieben werden und *online* Katalogen, die über das Internet verfügbar sind. In-house Systeme werden überall dort verwendet, wo der Kunde Geheimhaltung benötigt.

Dies kann daran liegen, dass er auch intern synthetisierte Substanzen in diesem Katalog listen möchte, für die unter Umständen noch kein Patentschutz vorliegt, dass der Katalog speziell verhandelte Preise oder Rabatte enthält oder auch daran, dass bereits die Information, *wonach* gesucht wird, für einen Konkurrenten interessant ist. Insbesondere bei Forschungsprojekten kann ein Konkurrent aus der Beobachtung der begleitenden Bestellungen und der Patentrecherchen auf das Ziel der Forschung

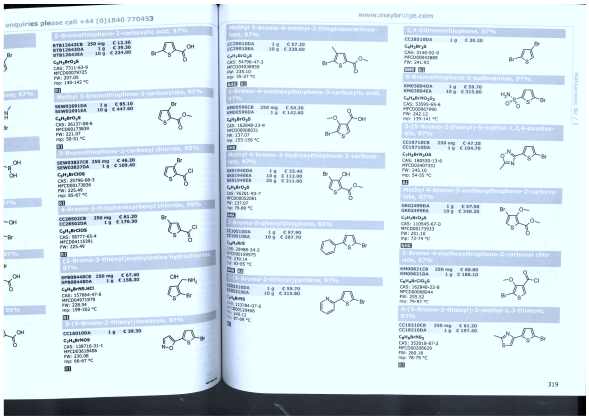


Abbildung II.3: Chemikalienkatalog in gedruckter Form - Maybridge Reference Handbook
Quelle: Maybridge [May08a]

schließen und sich so einen Wettbewerbsvorteil verschaffen. In-house Systeme können außerdem direkt an ein Warenwirtschaftssystem gekoppelt werden, so dass zumindest ein Teil der Bestellungen automatisch ausgeführt werden kann.

Eine besondere Klasse von in-house Katalogen sind Labor/Lager Einheiten, die als interne Kataloge, d.h. ohne Angabe von Bestandsmengen auftreten, wenn sie spezielle Chemikalien anbieten, die gegebenenfalls erst auf Anforderung hergestellt werden. Dies ist insbesondere notwendig, wenn es sich um *hazardous chemistry* handelt, die spezielle Syntheseanlagen und speziell geschultes Personal benötigt und/oder schwierig zu transportieren ist.

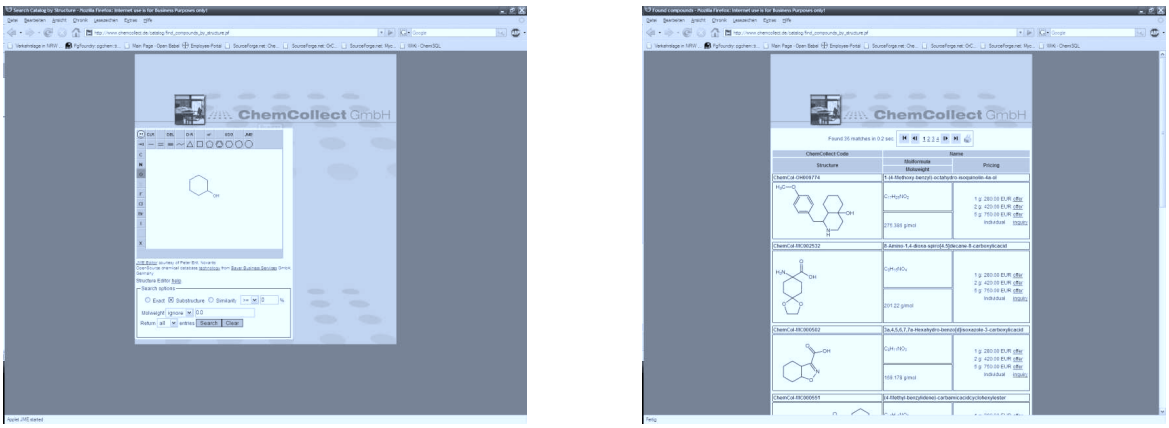


Abbildung II.4: Online-Chemikalienkatalog der Chemcollect GmbH
Quelle: Chemcollect GmbH

Online Kataloge dagegen stehen prinzipiell öffentlich zur Verfügung, wobei es sich in der Regel nicht um Shopsysteme mit direkter Bestellmöglichkeit handelt. Es ist zwar in einigen Fällen möglich, direkt aus dem System Bestellungen zu tätigen, diese werden aber nicht automatisch ausgeführt, sondern führen nur zu einer Bestellanforderung, die manuell weiterverarbeitet wird, da Chemikalien üblicherweise vielfältigen rechtlichen Restriktionen unterliegen, die auch noch von Legislative zu Legislative variieren und zudem vom Status des Bestellers (Privatperson, Unternehmen, Forschung & Lehre et cetera) abhängen. Abbildung II.4 zeigt die Suchmaske und das Suchergebnis eines solchen online Kataloges, Abbildung II.5 zeigt die Suchmaske und das Suchergebnis eines in-house Kataloges der Bayer Business Services GmbH (BBS).

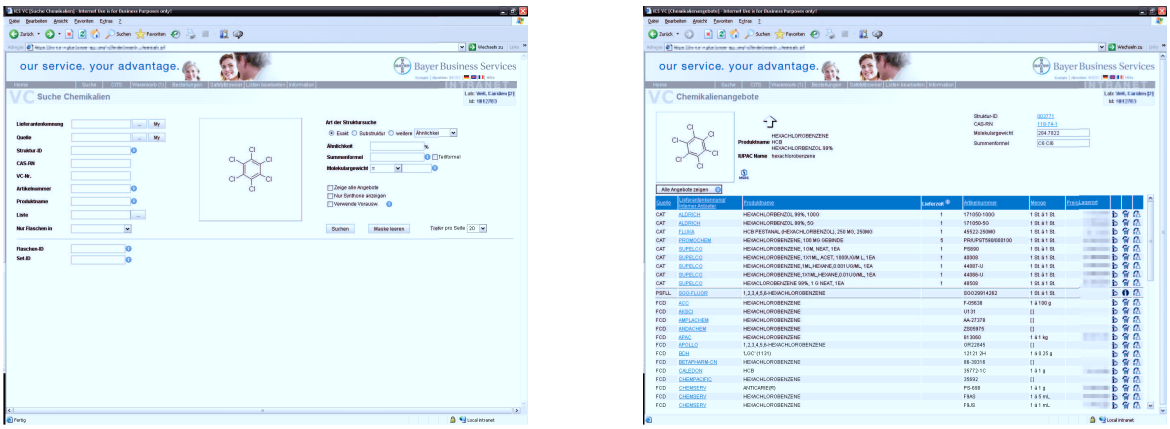


Abbildung II.5: In-house Chemikalienkatalog der BBS
Quelle: Bayer Business Services GmbH

In-house und online Kataloge für Chemikalien unterscheiden sich also hauptsächlich in der Zugriffskontrolle und dem Grad der Integration in den Beschaffungsprozess des sie nutzenden Unternehmens. Sie unterscheiden sich typischerweise nicht in der verwendeten Suchtechnologie und ihrer technischen Realisation. Die angebotenen Produkte sind für private Konsumenten weitgehend uninteressant. Beide sind mithin der Business-to-Business (B2B) Stufe im E-Procurement-Prozess innerhalb des E-Business zuzuordnen. Der Realisation der E-Business-Potentiale zur Verbesserung von Qualität und Quantität in der Informationsphase (vgl. [Dor01, S. 195-196]) stehen dabei die in Kapitel III beschriebenen Schwierigkeiten bei der Abbildung chemischer Strukturen in elektronischen Informationssystemen entgegen.

II.2.2 Patentrecherchesysteme

Ein weiteres zentrales Einsatzgebiet für strukturbasierte Suchen sind Patentrecherchesysteme, da für den Forschungsprozess der chemischen Industrie wesentliche Patente normalerweise ebenfalls über graphische Strukturinformationen identifiziert werden.

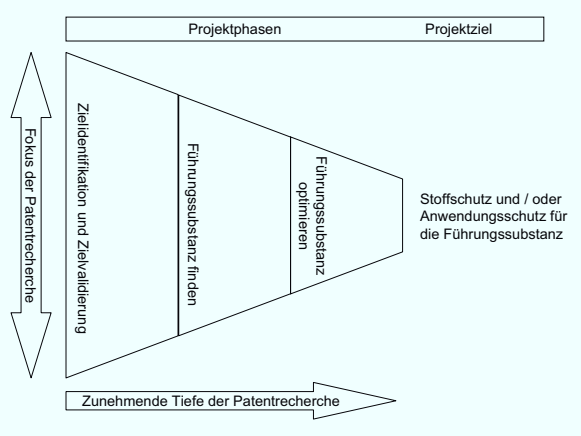


Abbildung II.6: Projektbegleitende Patentrecherche während eines Drug Discovery Projektes
Quelle: Eigene Darstellung

Patentrecherchen werden kontinuierlich während aller Stufen eines Drug Discovery Projektes durchgeführt, wobei, wie in Abbildung II.6 gezeigt, die Suchbreite ab- und die Suchtiefe zunimmt, je konkreter die Forschungsergebnisse werden.

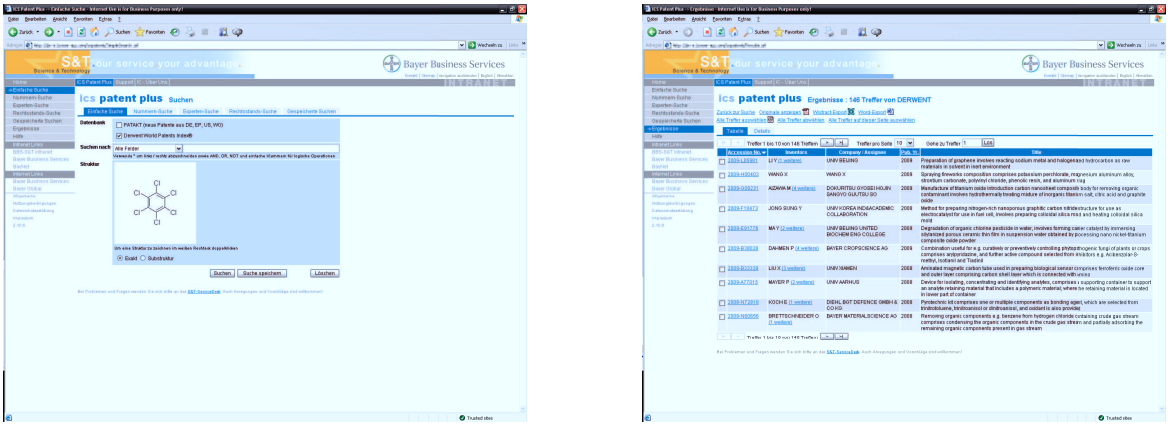


Abbildung II.7: In-house Patentrecherchesystem der BBS
Quelle: Bayer Business Services GmbH

Ziel ist es, die Patentsituation permanent zu analysieren, um mögliche Patentverletzungen frühzeitig zu erkennen und den eigenen Patentantrag so zu formulieren, dass möglichst der Stoff, mindestens aber die Anwendung oder Mischung durch das erteilte Patent geschützt werden.

Analog zu den Katalogsystemen sind Patentrecherchesysteme sowohl in-house (Abbildung II.7) als auch online, zum Beispiel über den Derwent Innovations Index [Der09], verfügbar.

II.2.3 Lagerverwaltungssysteme

Innerhalb des Unternehmens besteht die Notwendigkeit, extern beschaffte und selbst synthetisierte Chemikalien zwischenzulagern. Die Lagerung dient dabei vorrangig dem Zweck, ungenutzte Mengen für zukünftige Verwendung vorzuhalten und einen internen Beschaffungsmarkt zu etablieren, aus dem die Labore ihre Bedarfe unter Umständen synergistisch ohne externe Beschaffung decken können.

Ob in einem Labor oder einem dedizierten Lager gelagert wird, hängt dabei primär von Menge und Art der zu lagernden Substanz ab. Es gibt dabei sowohl aus den Substanzeigenschaften entstehende (zum Beispiel Geruch, Toxizität, Temperatursensibilität) als auch gesetzlich vorgegebene Lagerungsrestriktionen (zum Beispiel die maximale Menge an hochentzündlichen Substanzen, die in einem Labor gelagert werden darf), die nur durch spezielle Chemikalienläger erfüllt werden können.

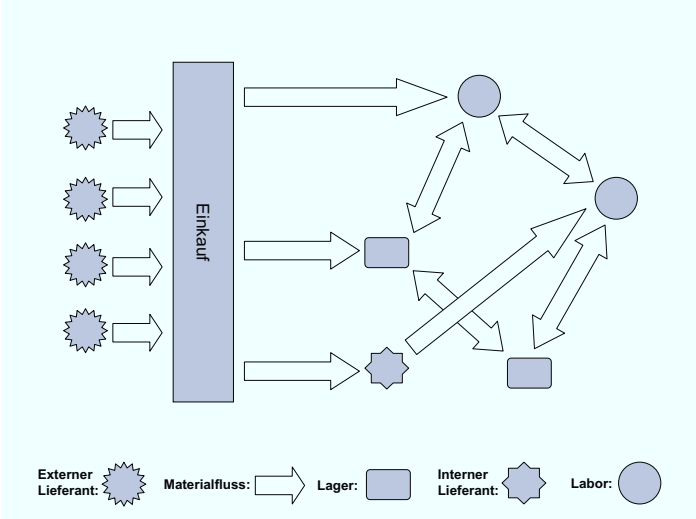


Abbildung II.8: Materialflüsse zwischen Laboren, Lägern, externen und internen Lieferanten
Quelle: Eigene Darstellung

Ein Lagerverwaltungssystem für Chemikalien muss also alle in Abbildung II.8 dem Einkauf nachfolgenden Objekte und Materialflüsse abbilden und verwalten können. Zusätzlich werden periodisch Reports erstellt, um gesetzliche Anforderungen, wie sie zum Beispiel aus dem deutschen Betäubungsmittelgesetz (BtMG) resultieren, zu erfüllen. Auch hier ist die Konformationsformel das wichtigste identifizierende Merkmal.

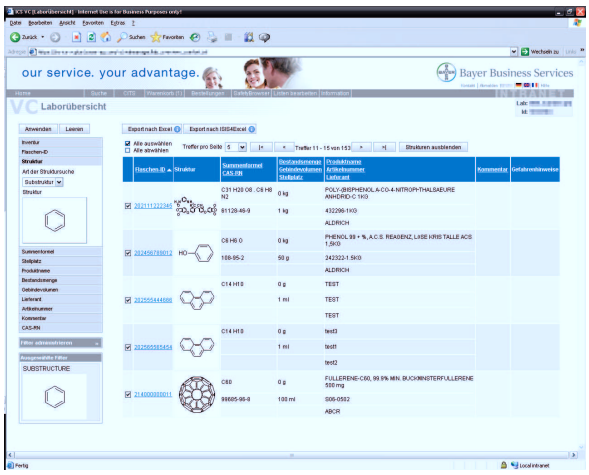


Abbildung II.9: Lagerverwaltungssystem für Chemikalien der BBS
Quelle: Bayer Business Services GmbH

Abbildung II.9 zeigt ein Suchergebnis in einem Lagerverwaltungssystem für Chemikalien der BBS.

II.2.4 Laborjournalsysteme

Ein Laborjournal ist ein Dokument, in dem der Fortgang des Forschungsprozesses, insbesondere die Planung, Durchführung und Ergebnisse wissenschaftlicher Experimente, aber auch Ideen für zukünftige Forschungsvorhaben, festgehalten werden. Das Laborjournal dient sowohl bei Patentstreitigkeiten gemäß § 3 Patentgesetz (PatG) als auch der Bewertung von Erfindungen gemäß § 9 Abs. 1 und 2 und § 10 Abs. 1 Gesetz über Arbeitnehmererfindungen (ArbnErfG) als Beweis für den genauen Zeitpunkt der Erfindung und die beteiligten Personen. Daher unterliegt die Führung eines Laborjournals strikten Regeln. Diese Regeln variieren zwar etwas je nach Institution, jedoch können folgende Punkte als Mindestanforderungen angenommen werden:

1. Das Laborjournal ist ein gebundenes Buch
2. Einträge werden mit dokumentenechter Tinte vorgenommen
3. Änderungen erfolgen durch Durchstreichen und Neueintrag der geänderten Stelle
4. Das Laborjournal wird zeitnah zum Experiment geführt
5. Rohdaten werden, wenn möglich, direkt im Laborjournal abgelegt

6. Einträge im Laborjournal werden von einem unabhängigen, fachkundigen Zeugen auf Vollständigkeit und Nachvollziehbarkeit geprüft und unterschrieben

Papierbasierte Laborjournale werden in zunehmendem Maße durch computergestützte Systeme, so genannte Electronic Lab Notebook (ELN), ersetzt. Insbesondere die Erfüllung von Punkt 5 wird dadurch wesentlich erleichtert. Teilweise übernehmen ELN bereits die Rohdaten direkt aus den Analysegeräten, so dass die manuelle Übertragung als Fehlerquelle weitgehend ausgeschlossen wird.

Weitere Vorteile von ELN sind die bessere Durchsuchbarkeit bereits dokumentierter Experimente sowie die automatische Speicherung großer Rohdatenmengen, wie sie zum Beispiel beim High-Throughput-Screening anfallen.

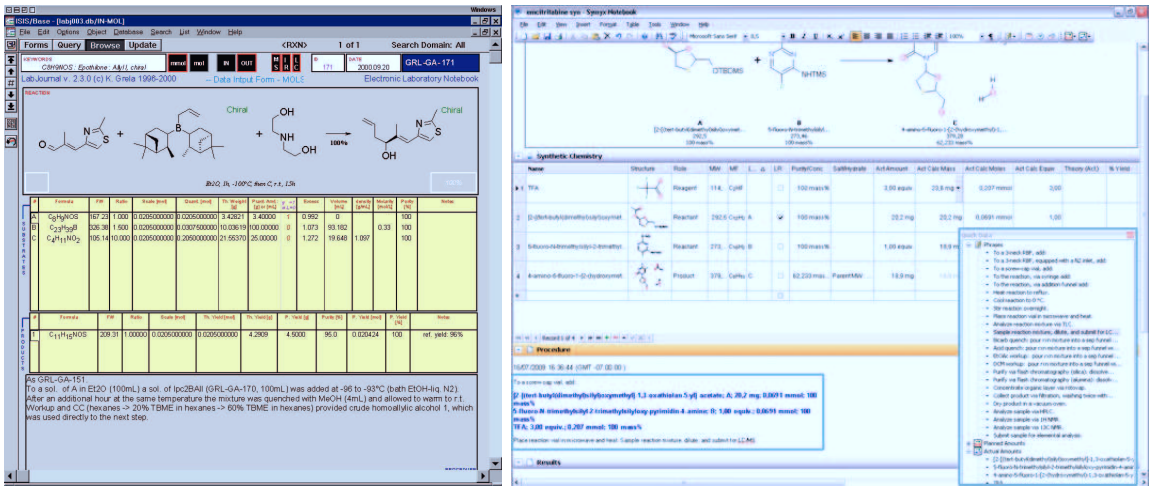


Abbildung II.10: Die ELN LabJ und Symyx Notebook
Quelle: [Gre09] und [SS09]

ELN müssen den Anforderungen des ISO/IEC 17025 [Int00] Standards für die elektronische Datenspeicherung und -sicherung genügen. Zusätzlich gelten die Anforderungen der jeweiligen Zulassungsbehörde, wenn das ELN in einem regulierten Umfeld, zum Beispiel der Arzneimittelforschung und -entwicklung, eingesetzt wird.

Abbildung II.10 zeigt das frei verfügbare ELN LabJ und das kommerzielle ELN Symyx Notebook for Synthesis.

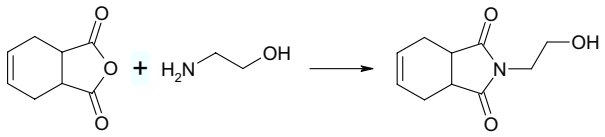


Abbildung II.11: Graphische Reaktionsformel
Quelle: Eigene Darstellung

In ELN ist die aus Strukturformeln und Symbolen zusammengesetzte graphische Reaktionsformel (Abbildung II.11) eines der wichtigsten identifizierenden Merkmale.

II.2.5 Anforderungen an chemische Informationssysteme

Aufgrund des, später in Unterabschnitt III.1.3 noch vertieft betrachteten, schlechten Laufzeitverhaltens der Algorithmen zur Erkennung von Graphen, ist die möglichst hohe Verarbeitungsgeschwindigkeit von Anfragen meist die zentrale Anforderung an chemische Informationssysteme.

Definition 1. *Der* Anfragedurchsatz D_q *ist die Menge an Anfragen* Q*, die ein System in einer festgelegten Zeit* t *verarbeiten kann.*

$$D_q = \frac{Q}{t} = \frac{t}{t_q}$$

Bei gegebenem t *ist* D_q *also umgekehrt proportional zu der für die Verarbeitung einer einzelnen Anfrage benötigten Zeit* t_q*.*

Definition 1 führt daher für alle gezeigte Anwendungsgebiete zu folgenden Anforderungen an die solchen Systemen zu Grunde liegenden Datenbanksysteme:

1. Es sind kurze Suchzeiten t_q je Suche gefordert, um akzeptable individuelle Antwortzeiten sowie einen guten Anfragedurchsatz D_q zu gewährleisten

2. Es wird eine möglichst geringe Streuung der Suchzeiten t_q über alle möglichen Suchen gefordert, um dem Anwender ein deterministisches Verhalten der Anwendung zu bieten und zu verhindern, dass Anfragen D_q überproportional beeinträchtigen oder das System sogar blockieren können (Denial Of Service (DoS))

Die Definition „akzeptabler“ individueller Antwortzeiten ist insofern schwierig, da die individuelle Akzeptanz ein Kriterium ist, welches stark vom jeweiligen Benutzer sowie der Art der Mensch-Maschine-Kommunikation abhängt. Als Richtlinie kann aber die von Miller publizierte Untersuchung über die maximal akzeptablen Antwortzeiten in der dialogischen Mensch/Computer Kommunikation „Response time in man-computer conversational transactions“ dienen. Miller basiert seine Empfehlungen dabei auf Erkenntnissen aus der Psychologie, die zeigen, dass Menschen Probleme um so weniger effizient lösen, je länger der Problemlösungsprozess, zum Beispiel durch Wartezeiten, unterbrochen wird.

Alle hier beschriebenen Systeme führen im Dialog mit dem Benutzer komplexe Abfragen aus und liefern Ergebnisse in Tabellen- und/oder graphischer Form. Miller benennt hierfür vier Sekunden als Höchstgrenze für Ergebnisse in Tabellenform [Mil68, S. 273] und zehn Sekunden als Höchstgrenze für Ergebnisse in graphischer Form [Mil68, S. 275], wenn der Gedankengang des Benutzers nicht unterbrochen werden soll. Bei mehr als 15 Sekunden Antwortzeit liegt aus Sicht des Benutzers kein Dialog mit dem System mehr vor: „In any event, response delays of approximately 15 seconds, and certainly any delays longer than this, rule out *conversational* interaction between human and information systems.“ [Mil68, S. 277]

Diese Anforderungen sind grundsätzlich für alle Informationssysteme anwendbar, haben aber für chemische Informationssysteme eine besondere Bedeutung, da hier aufgrund der graphischen Natur der gespeicherten Daten einige in Kapitel III genauer vorgestellte spezifische Probleme auftreten - die für andere Datentypen wie Zahlen oder Text heute praktisch als gelöst betrachtet werden können - an deren Lösung noch aktiv geforscht wird.

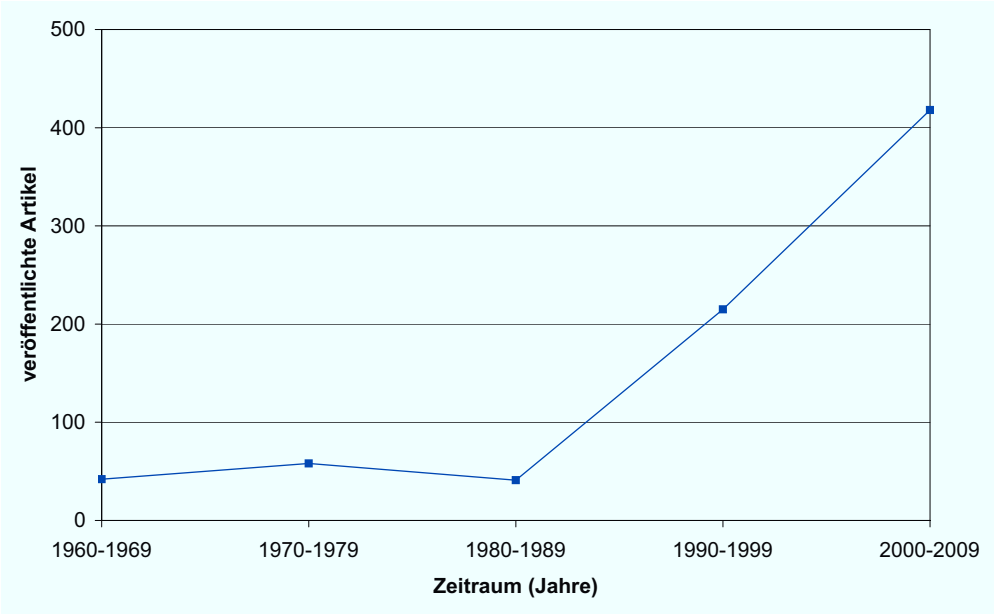


Abbildung II.12: Im *Journal of Chemical Information and Modeling* zu den Themen „Substruktursuche“, „Fingerprints“, „Ähnlichkeitsmaße“, „Datenbanken“, „Indexierung“ und „Clustering“ erschienene Artikel
Quelle: Eigene Darstellung

Als Indikator für diesen Trend dient exemplarisch der in Abbildung II.12 gezeigte Zuwachs an Publikationen zu diesem Themengebiet im *Journal of Chemical Information and Modeling* der American Chemical Society (ACS), welcher seit 1990 einen deutlichen positiven Trend zeigt. Die verwendete Suche zeigt Anhang I.

II.3 Informationsmanagement in der forschenden Chemieindustrie

Für den Begriff des Informationsmanagements existieren verschiedene Definitionen, zum Beispiel der *leitungszentrierte Ansatz* von Heinrich und Lehner:

„Mit dem Konstrukt *Informationsmanagement* wird also das Leitungshandeln (das Management) in einem Unternehmen in bezug auf Information und Kommunikation

bezeichnet, folglich alle Führungsaufgaben, die sich mit Information und Kommunikation im Unternehmen befassen. [...] Führungsaufgaben der Informationsfunktion sind insbesondere die Aufgaben der Schaffung, Nutzung und Weiterentwicklung ihrer Infrastruktur, kurz gesagt der **Informationsinfrastruktur**“ [LJH05, S. 7-8]

der *betriebswirtschaftliche Ansatz* von Brenner:

„Informationsmanagement ist eine *betriebswirtschaftliche Aufgabe*, wie Marketing oder Finanz- und Rechnungswesen. Seine Aufgabe ist es, betriebliche Zielsetzungen der Unternehmensführung zu erkennen, diese mit den Möglichkeiten der Informationstechnik zu kombinieren und computerunterstützte Anwendungen sowie organisatorische Lösungen zu entwickeln.“ [Bre07, S. 6]

oder der der *Information Resource Management (IRM)-Ansatz* von Stahlknecht und Hasenkamp:

„

a) primär die Aufgabe , den für das Unternehmen (nach Kapital und Arbeit) „dritten Produktionsfaktor“ *Information* zu beschaffen und in einer geeigneten *Informationsinfrastruktur* bereitzustellen, und

b) davon ausgehend die Aufgabe, die dafür erforderliche *IT-Infrastruktur*, d.h. die informationstechnischen und personellen Ressourcen für die Informationsbereitstellung

   - langfristig zu planen und
   - mittel- bis kurzfristig zu beschaffen und einzusetzen.

„ [SH05, S. 437]

Quintessenz aller Definitionen ist, dass die Aufgabe des Informationsmanagements die Schaffung einer geeigneten *Informationsinfrastruktur* innerhalb einer Organisation ist, welche diese Organisation bei der Erfüllung ihrer Aufgaben und Erreichung ihrer Ziele optimal unterstützt.

Die das Informationsmanagement für die Forschungsbereiche eines Unternehmens der Chemieindustrie definierenden Bedingungen sind:

1. Heterogene Hardware- und Softwarelandschaft

2. Datenaustausch zwischen Softwaresystemen und Messgeräten

3. Gesetzlich regulierte Umgebungen (Good Laboratory Practice (GLP), Good Manufacturing Practice (GMP), Good Clinical Practice (GCP))

4. Verarbeitung komplexer graphischer Datentypen wie Molekül, Reaktion und Spektrum in Softwaresystemen

5. Absolute Geheimhaltung des Forschungsprozesses bis zur Patenterteilung

Insbesondere Punkt 5 führt zu einer Besonderheit im Informationsmanagement der Forschungsbereiche eines Unternehmens der Chemieindustrie: Es zeigt eine bemerkenswerte Resistenz gegenüber aktuellen Trends wie dem Outsourcing von Informationstechnologie (IT) Dienstleistungen an Application Service Provider (ASP) oder Software as a Service (SaaS).

Dabei ist Outsourcing angesichts steigender Kosten auch im FuE-Bereich ein akzeptierter Versuch der Kostensenkung. Insbesondere die für die Zulassung eines Wirkstoffs vorgeschriebenen Studien werden regelmäßig an externe Dienstleister (Clinical Research Organisation (CRO)) vergeben, die mit deren Komplexität vertraut sind.

Das geschätzte Marktvolumen für CRO Dienstleistungen lag 2007 bei 17,8 Milliarden USD bei einer geschätzten Wachstumsrate von 15 % p.a. [ACR09]. Auch toxikologische Analysen, Verpackungs- und Formulierungsstudien werden regelmäßig extern vergeben.

Es lässt sich also feststellen, dass alle Schritte des FuE-Prozesses, die der erfolgreichen Patentierung *nachgelagert* sind, mögliche Kandidaten für das Outsourcing sind.

Alle Schritte *vor* der erfolgreichen Patentierung jedoch werden als *Kernkompetenz* betrachtet und unterliegen der *Geheimhaltung*, ebenso wie die zu ihrer Unterstützung und Verwaltung der Forschungsdaten benötigten in Abschnitt II.2 beschriebenen IT-Systeme. Dies hat zwei Gründe:

1. Es ist quasi unmöglich, den Umgang eines ASP mit digitalen Daten zu überwachen und gegebenenfalls zu sanktionieren, sobald diese das eigene Unternehmen verlassen haben: „And though „non malicious“ and trusted behavior of the ASP can be enforced by a contract, in many cases their legal power might be not sufficient. Differences in legal systems, security breaches in the ASP's infrastructure, finally, a virtual inability of the data owner to detect and prevent any misuse of

the outsourced data limit the applicability of a legislative regulation.“ [Evd08, S. 2]

2. Allein schon die Information was wann von wem *gesucht* wird, erlaubt einem kundigen Beobachter Rückschlüsse auf Ziel und Fortschritt des Forschungsprozesses zu ziehen

Daher werden selbst öffentlich im Internet verfügbare Informationen, wie Patentdatenbanken und Chemikalienkataloge, in In-house Systeme repliziert oder über anonymisierende Proxy-Server bereitgestellt. Die Replikation erlaubt dabei zusätzlich eine vereinfachte Integration von öffentlichen und geheimen Daten, ohne Systemgrenzen überschreiten zu müssen.

Kurz- bis mittelfristig liegt also das Hauptverbesserungspotential für das Informationsmanagement für die Forschungsbereiche eines Unternehmens der Chemieindustrie in der Optimierung der internen Systeme und Prozesse.

III Spezifische Probleme der Verwaltung chemischer Datentypen mit datenbankgestützten Informationssystemen und Ansätze zu deren Lösung

One of the main problems in creating a substructure-searchable chemical database is implementing the substructure search capability itself. This one requirement has done more to stifle the free flow of chemical information than perhaps any other. Solving the problem appears very difficult on first or second glance, and it is very difficult if you don't have the right tools. Many companies offer solutions - but at a price, both in terms of money and time, that is simply out of reach. - *R. Apocada*

III.1 Speicherung und Suche chemischer Konformationsformeln in relationalen Datenbanksystemen

III.1.1 Speicherung

Chemische Konformationsformeln werden typischerweise als ungerichtete, gewichtete Graphen $G = (V, E)$ repräsentiert. Die Gewichte der Knoten V sind dabei die notwendigen Informationen über das durch diesen Knoten repräsentierte Atom, die Gewichte der Kanten E die notwendigen Informationen über die durch diese Kante repräsentierte Bindung zwischen zwei Atomen. Einen solchen Graphen bezeichnet man als *Strukturgraphen*. Strukturgraphen müssen keine räumlichen Koordinaten enthalten, diese dienen nur der besseren Darstellung für den menschlichen Betrachter.

Es gibt diverse offengelegte Formate zur Speicherung solcher Strukturgraphen. Abbildung III.1 zeigt ein Beispiel des V2000 Molfile Formats. Die genaue Spezifikation dieses Textformats ist in [Sym07] beschrieben.

```
  -ISIS-  04210914102D

  4  3  0  0  0  0  0  0  0  0999 V2000
   -2.9042    0.7500    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   -2.1897    1.1625    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.4752    0.7500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.7608    1.1625    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
  2  3  2  0  0  0  0
  1  2  1  0  0  0  0
  3  4  1  0  0  0  0
M  END
```

Abbildung III.1: Symyx V2000 Molfile für (E)-2-Aminoethenol
Quelle: Eigene Darstellung

III.1.2 Suche

Die Zeit für die Verarbeitung einer Abfrage durch ein Datenbanksystem setzt sich aus zwei Faktoren zusammen: der Zeit für den Datenzugriff (I/O-Kosten) und der Zeit für die Weiterverarbeitung der Daten und die Präsentation des Ergebnisses (Berechnungskosten). „The cost of a DBMS query consists of two factors: I/O cost, the time spent in loading the data from the secondary storage into the main memory, and computational cost, the time spent by the DBMS in processing this data and returning the result.“ [BSAA04, S. 1021]

Typischerweise wird versucht, die Berechnungskosten durch den tatsächlichen Vergleich komplexer Datentypen, auch als *Sekundärfilter* bezeichnet, zu minimieren, indem man einen so genannten *Primärfilter* vorschaltet, welcher mittels einer weniger rechenintensiven Methode die Menge potentieller Kandidaten soweit einschränkt, dass der Sekundärfilter möglichst wenig unnötige Daten verarbeiten muss.

Kanth und Ravada haben gezeigt, dass für den Vergleich geographischer Datentypen in einem geographischen Informationssystem „the secondary-filter time is at least twice that of the filter time and dominates the overall computation time.“ [KR01, S. 406] Geographische Informationssysteme benutzen daher im Primärfilter vereinfachte Geometrien ihrer Inhalte, zum Beispiel deren Minimum Bounding Rectangle (MBR) oder Minimum Bounding Box (MBB) (vgl. [BSAA04]).

Prinzipiell ist die Primärfilterung in einem chemischen Informationssystem vergleichbar aufgebaut, allerdings können zum Beispiel keine Koordinaten für die Bestimmung einer MBB verwendet werden, da diese wie bereits in Unterabschnitt III.1.1 gezeigt, keine notwendigerweise im Strukturgraphen vorhandene Information sind.

Zusätzlich gibt es weitere spezielle Anforderungen an die Erkennung chemischer Graphen, die das Suchverfahren eines chemischen Informationssystems gegenüber einem geographischen Informationssystems weiter verkomplizieren. Beispiele solcher speziellen Anforderungen sind in Anhang C dargestellt.

III.1.3 Sekundärfilterung von Strukturgraphen

Substrukturerkennung

Die Substrukturerkennung stellt fest, ob eine Struktur Substruktur einer anderen ist, das heißt der eine Strukturgraph in dem anderen vollständig enthalten ist.

Definition 2. *Das Problem* Subgraph-Isomorphismus *besteht darin, für zwei ungerichtete Graphen G und G' festzustellen, ob G' ein Teilgraph von G ist. Der Graph $G' = (V', E')$ ist dann ein Teilgraph von $G = (V, E)$, wenn es eine injektive Abbildung $f : V' \to V$ gibt, so dass $\{u, v\} \in E' \to \{f(u), f(v)\} \in E$.*

Das Problem Subgraph-Isomorphismus ist ein Problem in der Komplexitätsklasse nichtdeterministisch polynomiale Zeit (NP) und NP-vollständig (vgl. [GJ79]). Das bedeutet, dass alle bekannten Lösungsalgorithmen nicht-polynomiale Laufzeiten[1] aufweisen.

Exakte Strukturerkennung

Die exakte Strukturerkennung stellt fest, ob zwei Strukturen gleich, das heißt ihre Strukturgraphen gleich sind.

Definition 3. *Das Problem* Graph-Isomorphismus *besteht darin, für zwei ungerichtete Graphen G und G' festzustellen, ob $G' \cong G$ ist. Der Graph $G' = (V', E')$ ist dann isomorph zu $G = (V, E)$, wenn es eine bijektive Abbildung $f : V' \to V$ gibt, so dass $\{u, v\} \in E' \leftrightarrow \{f(u), f(v)\} \in E$.*

Die Erkennung des Graph-Isomorphismus ist ein Problem in der Komplexitätsklasse NP, es ist aber weder bekannt, ob es eine Lösung mit polynomialer Laufzeit gibt, noch ob

[1] Das heisst ein Laufzeitverhalten von $\mathcal{O}(n^k)$, wobei $k > 1$ eine Konstante und n die Größe der Eingabemenge ist.

es NP-vollständig ist (vgl. [GJ79]). Das bedeutet, die Frage seiner Laufzeitkomplexität ist bisher ungelöst.

Das Problem lässt sich allerdings umgehen, indem Hashwerte[2] aus den Strukturgraphen berechnet und mit abgespeichert werden. Dann reduziert sich die obere Grenze des Laufzeitverhaltens auf $\mathcal{O}(n)$ für den elementweisen Vergleich zweier Hashwerte mit n Elementen. Ein geeigneter Hashwert mit sehr geringer Kollisionswahrscheinlichkeit ist beispielsweise der InChIKey: „For duplication of only the first block of 14 characters this [die Kollisionswahrscheinlichkeit; Anm. d. Verf.] is 1.3% in 10^9, equivalent to a single collision in one of 75 databases of 10^9 compounds each.“ [Int07] Falls nötig lässt sich die Kollisionswahrscheinlichkeit durch die Kombination mehrerer, durch verschiedene Hashfunktionen erzeugter Hashwerte weiter reduzieren.

III.1.4 Primärfilterung (Screening) mittels Deskriptoren und Deskriptorenvektoren

Wie in Unterabschnitt III.1.2 gezeigt, ist das Laufzeitverhalten einer vollständigen Überprüfung von Strukturgraphen auf Subgraphen-Isomorphismus orthogonal zur Anforderung möglichst schneller Suchen in einem Strukturdatenbestand. Um trotzdem akzeptable Suchzeiten zu erreichen, wird das in Unterpunkt III.1.2 bereits vorgestellte Konzept der Primärfilterung in Form eines geeigneten Index auch auf den Datentyp des Strukturgraphen angewandt. Hierzu muss zunächst der Begriff des *Molekül- oder Strukturdeskriptors* definiert werden.

Definition 4. *„Molecular descriptors create numeric values for a mathematical characterization of the structure and the environment of a molecule.“ [Kuh09, S. 37]*

Speichert man einen solchen Deskriptor für jeden Strukturgraphen im Datenbestand, ist es möglich, vor dem eigentlichen Vergleich der Strukturgraphen die Menge der möglichen Kandidaten einzuschränken wie in Abbildung III.2 schematisch gezeigt.

[2] Ein Hashwert ist das Ergebnis einer Hashfunktion. Die Hashfunktion bildet jeden Eingabewert aus einer Quellmenge auf eine (typischerweise kleinere) Menge möglicher Ausgabewerte ab. Der erzeugte Hashwert kann dann zum Beispiel als Schlüssel für den Eingabewert benutzt werden. Hashfunktionen sind nicht injektiv, dadurch besteht eine gewisse Wahrscheinlichkeit der Abbildung mehrerer Elemente der Quellmenge auf denselben Hashwert, die sogenannte *Kollisionswahrscheinlichkeit*.

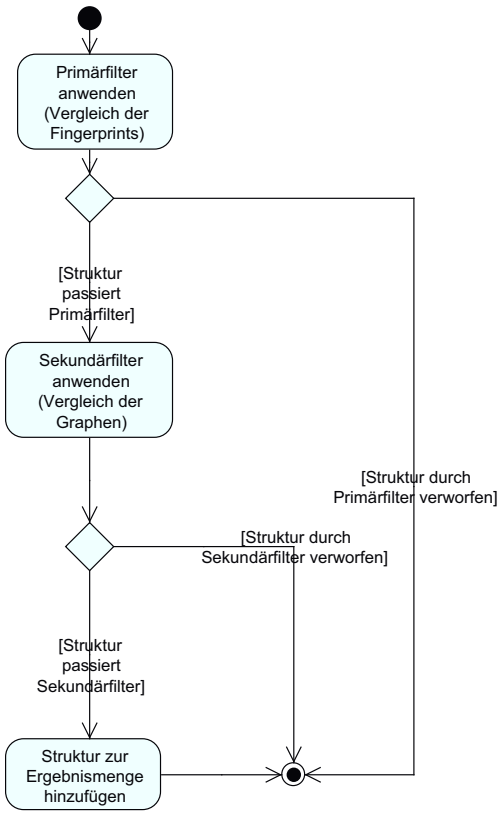


Abbildung III.2: Schematischer Ablauf einer Struktursuche mit Primär- und Sekundärfilterung
Quelle: Eigene Darstellung

Wurde zum Beispiel die Molekülmasse[3] vorberechnet und mit gespeichert, können bei einer Substruktursuche a priori alle Strukturen ausgeschlossen werden, deren molare Masse kleiner als die der gesuchten Struktur ist. Eine solche Primärfilterung bezeichnet man in chemischen Informationssystemen auch als *Screening*. Die Anforderungen an ein Screeningverfahren sind:

1. Es sollen möglichst wenige *false positives*, also Kandidaten, die den nachfolgenden Sekundärfilter nicht passieren, durchgelassen werden

2. Es dürfen keine *false negatives*, also Kandidaten, die den nachfolgenden Sekundärfilter passieren würden, ausgefiltert werden

[3] Die Molekülmasse einer chemischen Verbindung gibt an, wie groß die Masse eines Moleküls dieser Verbindung im Vergleich zu einem Zwölftel der Masse des häufigsten Kohlenstoffisotops ^{12}C ist.

3. Der im Screeningverfahren verwendete Primärfilter muss deutlich schneller durchzuführen sein als der Sekundärfilter, um die Berechnungskosten niedrig zu halten

4. Die im Screeningverfahren verwendeten Deskriptoren sollten weniger Speicherplatz benötigen als die Daten, aus denen sie berechnet wurden, um die I/O-Kosten niedrig zu halten

Der zusätzliche Aufwand für die Vorberechnung des Deskriptors fällt dabei nur einmalig beim Speichern einer Struktur an und kann daher für alle nachfolgenden Suchen vernachlässigt werden.

Ein bereits genanntes Beispiel für einen Strukturdeskriptor ist die Molekülmasse des Moleküls. Die Molekülmasse ist allerdings ein wenig selektiver Strukturdeskriptor, da sie keinerlei Konformationsinformationen enthält.

Daher wurde eine Vielzahl von Deskriptoren entwickelt, die sowohl Konformationsinformationen kodieren als auch eine kompakte Speicherung ermöglichen. Um eine Struktur möglichst genau zu beschreiben, werden mehrere solcher Deskriptoren zusammen in einem Deskriptorenvektor (III.1) zusammengefasst.

$$S \rightarrow \vec{S} \tag{III.1}$$

Typischerweise ist diese Abbildung verlustbehaftet, der Strukturgraph kann also nicht aus dem Deskriptorenvektor rekonstruiert werden. Deskriptorenvektoren können dann in einer für die Indexierung geeigneten Datenstruktur, zum Beispiel einer Liste oder einem Baum, gespeichert werden und bilden den Index.

Je nach Inhalt und Erzeugungsverfahren des Deskriptorenvektors unterscheidet man zwischen *structural statistics*, *structural keys* und *hashed path fingerprints*.

III.1.5 Structural statistics

Structural statistics kodieren sowohl die Existenz eines Merkmals als auch dessen Anzahl als $\vec{S} = \begin{pmatrix} S_1 & S_2 & \dots & S_i \end{pmatrix}$ $(S_i \in \mathbb{N}_0)$. Dabei bedeutet $S_i > 0$, dass das Merkmal i genau S_i mal vorkommt, $S_i = 0$, dass es nicht vorkommt. Das Merkmal selbst wird durch die Position des Eintrags im Deskriptorenvektor identifiziert.

Das Programmpaket CHECKMOL / MATCHMOL [Hai09a] kann solche Vektoren erzeugen und das darauf basierende Strukturdatenbanksystem [Hai09b] sowie die ersten Versionen von PGCHEM::TIGRESS verwenden eine wie in Listing III.1 gezeigte Datenbanktabelle mit den vorberechneten *structural statistics* als Index auf die gespeicherten Strukturen.

```
CREATE TABLE structstats
(
  iiid int4 NOT NULL DEFAULT 0,
  n_atoms int2 NOT NULL DEFAULT 0,
  n_bonds int2 NOT NULL DEFAULT 0,
  n_rings int2 NOT NULL DEFAULT 0,
  n_qa int2 NOT NULL DEFAULT 0,
  n_qb int2 NOT NULL DEFAULT 0,
  n_chg int2 NOT NULL DEFAULT 0,
  n_c1 int2 NOT NULL DEFAULT 0,
  ...
  CONSTRAINT structstats_pkey PRIMARY KEY (iiid)
  )
```

Listing III.1: Datenbanktabelle zur Speicherung von structural statistics
Quelle: Eigene Darstellung

Im Falle von CHECKMOL/MATCHMOL wird die Zuordnung von Deskriptor zu Position starr zur Übersetzungszeit festgelegt. Die Zuordnung könnte aber auch über ein extern konfigurierbares Wörterbuch mit den Substrukturmustern der gewünschten Deskriptoren erfolgen.

Die Hauptvorteile von structural statistics sind ihre einfache Integrierbarkeit in relationale Datenbanksysteme durch die Speicherung in Datenbanktabellen und die Verwendung von Structured Query Language (SQL) Ausdrücken für das Screening über solche Tabellen.

```
SELECT iiid FROM
structstats WHERE
n_atoms>=13 AND n_bonds>=15 AND
n_rings>=6 AND n_C2>=11 AND
n_C>=11 AND n_CHB1p>=4 AND
n_N2>=1 AND n_N3>=1 AND n_b2>=6 AND
n_bar>=15 AND n_CN>=4 AND
n_rN>=5 AND n_rN1>=3 AND n_rN2>=2 AND
n_rX>=5 AND n_rar>=6
```

Listing III.2: SQL für das Substruktur-Screening in structural statistics
Quelle: Eigene Darstellung

Hauptnachteile sind der vergleichsweise hohe Speicherplatzbedarf der ganzzahligen Datentypen, die Begrenzung der Anzahl möglicher Deskriptoren in $\vec{S}$ durch die maximale Spaltenzahl pro Tabelle des verwendeten RDBMS sowie die für die Abfrage notwendigen unhandlichen SQL Ausdrücke, wie das Beispiel in Listing III.2 verdeutlicht.

III.1.6 Structural keys

Structural keys benutzen ein Wörterbuch, um die korrespondierenden Bits in $\vec{S} = \begin{pmatrix} S_1 & S_2 & \dots & S_i \end{pmatrix}$ $(S_i \in 0, 1)$ zu setzen, indem mit jedem Eintrag im Wörterbuch eine Substruktursuche auf der Eingabestruktur durchgeführt wird und im Trefferfall ein oder mehrere Bits gesetzt werden.

Man unterscheidet weiterhin zwischen structural keys mit direkter und indirekter Deskriptorenkodierung. Je nach Implementierung sind structural keys anfällig für exzessive Selektivität und können Bedingung 2 der Anforderungen an die Screening-Methode aus Unterpunkt III.1.4 verletzen, wie in Anhang E gezeigt wird.

Structural keys mit einfacher Deskriptorenkodierung Jeder Deskriptor aus dem Wörterbuch setzt direkt das ihm zugeordnete Bit. Eine Kodierung weiterer Merkmale, zum Beispiel der Anzahl der Treffer des Suchargumentes in der Zielstruktur, findet nicht statt. Der OpenBabel[Ope09] FP3 Algorithmus ist ein Vetreter dieser Klasse. Hauptvorteile der einfachen Deskriptorenkodierung sind die exakte Steuerbarkeit der

Vektorbits durch ein Wörterbuch, die Bijektivität der Abbildung und die Kollisionsfreiheit, da ein Bit nicht von mehreren Deskriptoren gesetzt werden kann. Hauptnachteil ist, dass keine weiteren Informationen gespeichert werden können.

Structural keys mit erweiterter Deskriptorenkodierung Bei der erweiterten Deskriptorenkodierung können neben der Existenz eines Deskriptors in der Eingabestruktur weitere Merkmale, zum Beispiel die Anzahl seines Vorkommens, kodiert werden.

Die MACCS Keys der Firma Symyx benutzen eine interne Projektionstabelle. Ein Eintrag in dieser Projektionstabelle wird als *keybit definition* bezeichnet und ist folgendermaßen aufgebaut: „Specifically, a keybit is defined by nine numbers, which we will denote by n1 through n9. The first four numbers encode the various properties into descriptors. The remaining five numbers determine which keybits are set by the descriptor.“ [DLHN02, S. 1274]

Zum Beispiel bedeutet der Eintrag in Abbildung III.3: „At least two occurrences (n4) of an atom in a multiple, nonaromatic bond (n2) located two bonds (n1) away from an atom with at least two heteroatom neighbors (n3). The descriptor sets three keybits (n5), which can be set by other descriptors (n6). The keybits set are 479, 469, and 763 (n7-n9).“ [DLHN02, S. 1275]

position:	n1	n2	n3	n4	n5	n6	n7	n8	n9
key:	2	3	5	2	3	1	479	469	763

Abbildung III.3: Beispiel für eine *keybit definition*
Quelle: [DLHN02, S. 1275]

Zusätzlich zu den in Absatz III.1.6 genannten Vorteilen weisen diese Verfahren eine höhere Selektivität gegenüber der einfachen Positionskodierung aus und, im Fall der Symyx Keys die Möglichkeit, die Abbildung von Deskriptoren auf einzelne Bits über die Projektionstabelle zu beeinflussen.

Ihre Hauptnachteile bestehen im erhöhten Speicherplatzbedarf und, im Fall der Symyx Keys, der Notwendigkeit, neben dem Wörterbuch auch noch eine optimierte Projektionstabelle aufbauen zu müssen, da die Anzahl theoretisch möglicher *keybits* und *keybit*

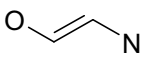
definitions doch recht unhandlich ist: „The existing MDL 2D keyset technology can, in theory, produce in excess of 3 million distinct keybits, which can be combined into innumerable keysets.“ [DLHN02, S. 1273]

Structural keys mit hybrider Deskriptorenkodierung Das PubChem System [Pub09b] benutzt einen hybriden Deskriptorenvektor (vgl. [Pub09a]), bei dem Existenzkodierung und Anzahlkodierung gemischt benutzt werden. Der in Anhang D beschriebene FPPC8 Algorithmus erzeugt ebenfalls hybride structural keys.

III.1.7 Hashed Path Fingerprints

Dieser Ansatz unterscheidet sich von den bisher gezeigten dadurch, dass die Deskriptoren aus dem Strukturgraphen nicht exogen, sondern endogen erzeugt werden. Es gibt also keine externe Vorschrift in Form eines Wörterbuchs, nach der die Bits im Deskriptorenvektor gesetzt werden. Der Algorithmus arbeitet vielmehr so, dass der Strukturgraph zunächst erschöpfend in Pfade bis zu n Bindungen Länge zerlegt wird, wie Tabelle III.1 zeigt.

Tabelle III.1: Mustererzeugung aus Pfaden eines Strukturgraphen
Quelle: Eigene Darstellung



Länge	**Pfad**		
1	C	O	N
2	O-C	C=C	C-N
3	O-C=C	C=C-N	
4	O-C=C-N		

Jeder dieser Pfade wird dann über eine Hashfunktion auf ein oder mehrere Bits in $\vec{S} = \begin{pmatrix} S_1 & S_2 & \dots & S_n \end{pmatrix}$ $(S_n \in 0, 1)$ abgebildet. Anders als bei exogen erzeugten De-

skriptorenvektoren kann man die Position eines gesetzten Bits nicht einem bestimmten Merkmal der Eingabestruktur zuordnen. Allerdings ist gewährleistet, dass gleiche Eingabestrukturen gleiche Pfade erzeugen und somit auch dieselben Bits setzen.

Die Hauptvorteile von hashed path fingerprints sind ihre grundsätzliche Eignung für alle Strukturgraphen, da sie, außer der maximalen Pfadlänge, keinerlei Annahmen über „interessante“ und „uninteressante“ Merkmale machen sowie die vergleichsweise hohe Geschwindigkeit des Algorithmus. Hauptnachteil ist die eingeschränkte Selektivität durch Hashkollisionen und *blinde Stellen*. Der in PGCHEM::TIGRESS verwendete OPENBABEL FP2 Algorithmus zeigt diese typischen Probleme, die in Abschnitt III.2 vertieft betrachtet werden.

III.2 Analyse und Formulierung der Problemstellung

PGCHEM::TIGRESS verwendet für die Indexierung von Molekülen einen domänenspezifischen Generalized Search Tree (GiST) Index wie in Anhang F, Abschnitt F.2 beschrieben. Die notwendigen Deskriptorenvektoren für die Indexeinträge werden primär durch den FP2 Algorithmus der OPENBABEL Bibliothek erzeugt.

Tabelle III.2: Mehrfach vorkommende Fingerprints in verschiedenen Chemikalienkatalogen
Quelle: Eigene Darstellung anhand von [Che09b] [May08b] [Asi08b] und [Asi08a]

Katalog	Jahr	Fingerprints	irrtümlich mehrfach	%
ChemCollect Katalog	2009	23 200	2 253	10
Maybridge Screening Compounds	2008	58 159	2 056	4
Asinex Platinum Collection	2008	125 231	17 932	14
Asinex Gold Collection	2008	229 398	24 749	11

Allerdings ist dieser Algorithmus anfällig dafür, für chemisch verschiedene Moleküle identische Bitmuster zu erzeugen, wie in Tabelle III.2 für die in dieser Arbeit verwendeten Kataloge kommerziell verfügbarer Chemikalien gezeigt wird. Dies hat zwei Ursachen:

1. Hashkollisionen (siehe Unterpunkt III.2.1)

2. *Blinde Stellen* aufgrund repetitiver Pfade im Molekülgraphen (siehe Unterpunkt III.2.2)

III.2.1 Hashkollisionen

Aufgrund des Schubfachprinzips von Dirichlet (Definition 5) ist ein hashed path fingerprint anfällig für Hashkollisionen.

Definition 5. *Dirichlet's Schubfachprinzip [Wei09] besagt, dass, wenn unter der Bedingung* $(n, m \in \mathbb{N};\ n > m)$ *n Objekte auf m Schubfächer verteilt werden, mindestens ein Schubfach mehr als ein Objekt enthalten muss.*

$$p(n) = 1 - \frac{m!}{(m-n)! \cdot m^n} = 1 \ (n, m \in \mathbb{N}\ n > m) \tag{III.2}$$

Dies folgt aus dem Spezialfall des Geburtstagsparadoxons (III.2) für $n > m$*: Die Wahrscheinlichkeit, dass mindestens zwei der n Objekte im selben Schubfach liegen, ist dann* $p(n) = 1$.

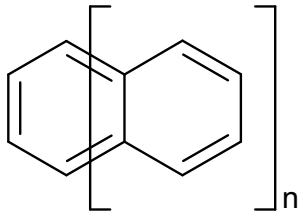
Für den FP2 Algorithmus ist $n = 1021$ und $m \to \infty$. Wie groß m genau ist, hängt von der Anzahl der unterscheidbaren Moleküle in der betrachteten Datenbasis ab, eine obere Grenze kann nur geschätzt werden. Für organische Verbindungen mit bis zu 30 Atomen der Elemente C, N, O und S schätzen Bohacek, McMartin und Guida bereits mehr als 10^{60} mögliche, stabile Moleküle [BMG96, S. 43]. Tabelle III.2 zeigt, dass bereits bei deutlich kleineren Datenmengen Kollisionen auftreten.

III.2.2 Blinde Stellen

Zusätzlich weist der Algorithmus *blinde Stellen* auf. Blinde Stellen entstehen, wenn für unterschiedliche Strukturen identische Pfade erzeugt werden, wie in Tabelle III.3 am Beispiel kondensierter homozyklischer, d.h., alle ringbildenden Atome stammen nur von einem chemischen Element, Ringe gezeigt wird. Für alle $n > 1$ werden repetitiv identische Pfadfragmente und somit identische Fingerprints erzeugt. Für eine Erklärung der Fragmentnotation siehe Anhang B.

Tabelle III.3: Fragmentgenerierung des FP2 Algorithmus für kondensierte Benzolringe
Quelle: Eigene Darstellung

Kondensierte Benzolringe



n	**Fragmente**	**Konfiguration**
> 1	0|6|5|6	linear
	0|6|5|6|5|6	linear
	0|6|5|6|5|6|5|6	linear
	0|6|5|6|5|6|5|6|5|6	linear
	0|6|5|6|5|6|5|6|5|6|5|6	linear
	5|6|5|6|5|6|5|6|5|6|5|6	zyklisch
	0|6|5|6|5|6|5|6|5|6|5|6|5|6	linear

Blinde Stellen des FP2 Algorithmus existieren mindestens noch für die kondensierten homozyklischen Ringe aus Abbildung III.4. Blinde Stellen führen zu false positives in der Screeningphase. Beispielsweise werden für eine Suche nach Anthracen auch alle Einträge, die nur Naphthalen enthalten, zurückgegeben und erst durch die folgende Überprüfung auf Subgraphen-Isomorphismus verworfen.

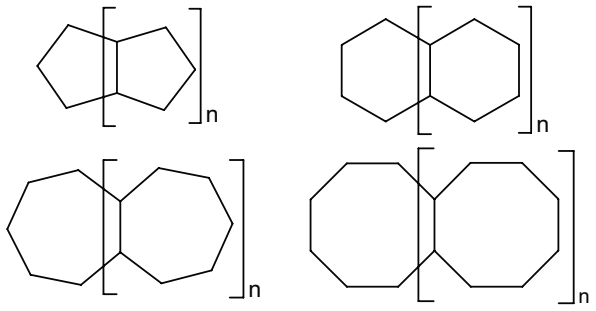


Abbildung III.4: Weitere kondensierte Ringe, für die der FP2 Algorithmus blinde Stellen erzeugt
Quelle: Eigene Darstellung

Ein entsprechend angepasster Index, bei dem zusätzlich zum FP2 Deskriptorenvektor ein FPPC8 Deskriptorenvektor verwendet wird, der Naphthalen explizit kodiert, verbessert die Suchzeiten signifikant, wie in Tabelle III.4 gezeigt.

Tabelle III.4: Mittlere Suchzeiten für Anthracen im Maybridge Screening Collection (MAYSC) Katalog
Quelle: Eigene Darstellung

Indexkonfiguration	**Ø Suchzeit in Sekunden**
ohne Index	110
mit Index, ohne optimierten Fingerprint	15
mit Index, mit optimiertem Fingerprint	0,5

Formulierung des Realproblems Daraus ergibt sich folgende Formulierung des zu lösenden *Realproblems* DICTIONARY:

Finde eine Kombination von Mustern $\binom{n}{k}$, welche die Selektivität des FPPC8 Fingerprints für einen gegebenen Datenbestand maximiert, wobei gilt: $k <= n$.

Die Menge aller zur Verfügung stehenden Substrukturmuster wird im Folgenden als *Basiswörterbuch*, eine Kombination $\binom{n}{k}$ als *reduziertes Wörterbuch* bezeichnet.

III.3 Mögliche Ansätze zur Lösung des Realproblems mit den Methoden des Operations Research

Methoden zur Lösung des in Abschnitt III.2 formulierten Realproblems sind in der Domäne des Operations Research (OR) zu finden.

Es gibt verschiedene Definitionen des Begriffs OR, zum Beispiel von Müller-Merbach:

„Es soll daher unter dem Begriff Optimalplanung [als Synonym für Operations Research; Anm. d. Verf.] die Anwendung von mathematischen Methoden zur Vorbereitung optimaler Entscheidungen verstanden werden."[MM92, S. 1]

oder Churchman, Ackoff und Arnoff:

„...als die Anwendung wissenschaftlicher Methoden und Verfahren auf Probleme betreffend die Arbeitsweise von Systemen, mit dem Ziel, jenen Personen, die diese Arbeitsweise lenken, optimale Lösungen für diese Probleme zu liefern."[CWC71, S.18]

Müller-Merbach zielt dabei mehr auf die *Vorbereitung optimaler Entscheidungen*, während Churchman, Ackoff und Arnoff *optimale Lösungen* von Problemen erwarten.

Gleich welche Definition man bevorzugt, lässt sich feststellen, dass OR eine Disziplin der angewandten Mathematik ist, welche sich mit der optimalen Lösung von Planungs- und Entscheidungsproblemen mittels mathematischer Methoden befasst.

Formulierung des Formalproblems Um solche Methoden anwenden zu können, muss das Realproblem Dictionary zunächst in ein *Formalproblem* [MM92, S.14 ff.] überführt werden.

Finde eine Kombination von Elementen $\binom{n}{k}$ mit einem individuellen Nutzen u und einem individuellen Gewicht g, so dass der Gesamtnutzen U maximiert wird, ohne das höchstzulässige Gewicht G zu überschreiten.

Dieses Formalproblem wird allgemein als Rucksackproblem Knapsack bezeichnet.

Bestimmung des Gesamtnutzens der Lösung Der Gesamtnutzen U einer Lösung kann durch die so genannte Fitnessfunktion bestimmt werden.

Definition 6. *Die Fitnessfunktion ϕ beschreibt die Fitness einer Lösung und setzt sich aus der Zielfunktion F und der Dekodierfunktion Γ zusammen.*

Die Dekodierfunktion $\Gamma(\vec{e}_j)$ ist hier die Anwendung eines reduzierten Wörterbuches aus k Elementen auf die Menge der false duplicates der Deskriptorenvektoren $\vec{e}_j$.

$$\phi(\vec{e}_j) = F(\Gamma(\vec{e}_j)) \tag{III.3}$$

Die Zielfunktion F ist die Veränderung der Suchzeiten mit den optimierten Deskriptorenvektoren bezogen auf die Suchzeiten mit den nicht optimierten Deskriptorenvektoren.

Da die Fitness der Lösung jedoch bei *jedem* Optimierungsschritt berechnet werden muss, um die Güte der aktuellen Lösung zu bewerten und ihr Ergebnis neben der angewandten Optimierungsmethode auch noch von den verwendeten Suchkriterien abhängt, wurde eine von diesem Faktor unabhängige und weniger laufzeitintensive Ersatzzielfunktion F' gewählt.

Dazu werden zunächst alle false duplicates der Deskriptorenvektoren in eine Quarantänetabelle T_Q kopiert.

F' ist dann die Verbesserung der Selektivität Δ_S durch $\vec{e}_j$, ausgedrückt als der Anteil eindeutiger Deskriptorenvektoren in der optimierten T_Q bezogen auf den Anteil eindeutiger Deskriptorenvektoren in der nicht optimierten T_Q, welcher per Definition immer gleich null ist.

$$\phi(\vec{e}_j) = F'(\Gamma(\vec{e}_j)) \tag{III.4}$$

In allen folgenden Optimierungsansätzen wird F' als Maß für den Gesamtnutzen U verwendet. In Unterabschnitt IV.3.2 wird dann ex post überprüft, inwieweit die mit Hilfe von F' gewonnenen Ergebnisse tatsächlich mit F korrelieren.

III.3.1 0-1-Knapsack

Da im reduzierten Wörterbuch jedes gleiche Element nur einmal vorkommen darf, weil mehrere gleiche Muster zu keinem weiteren Selektivitätsgewinn führen würden, handelt es sich bei DICTIONARY sogar um ein *binäres* Rucksackproblem 0-1-KNAPSACK der Form:

$$\begin{aligned} \text{Maximiere} & \sum_{j=1}^{n} u_j x_j \\ \text{unter den Bedingungen} & \sum_{j=1}^{n} g_j x_j \leq G \\ & x_j \in \{0,1\} \\ & j = \{1, \cdots, n\} \\ & x_j = \begin{cases} 1 \text{ wenn Element j ausgewählt} \\ 0 \text{ wenn Element j nicht ausgewählt} \end{cases} \end{aligned} \tag{III.5}$$

Das Problem 0-1-KNAPSACK ist NP-vollständig. Der diesbezügliche Beweis auf Basis der von KARP bewiesenen NP-Vollständigkeit des SUBSET-SUM Problems (vgl. [Kar72]) wird zum Beispiel in [MT90, S. 6] gezeigt.

Zusätzliche Bedingungen

Für DICTIONARY gelten noch weitere Bedingungen:

$$g_j = 1 \tag{III.6}$$

$$G \in \mathbb{N} \tag{III.7}$$

Bedingung III.6 resultiert daraus, dass das Gewicht jedes Elements g_j als identisch, ganzzahlig und konstant angenommen werden kann, da zwar ein potentielles variables Gewicht in Form der notwendigen Suchzeit für die Substruktursuche jedes Elements existiert, dieses aber nur zur Indexierzeit und nicht zur Suchzeit anfällt und somit zu

vernachlässigen ist. Ist das Gewicht g_j aller Einträge positiv ganzzahlig und konstant, kann der Einfachheit halber $g_j = 1$ angenommen werden.

Bedingung III.7 folgt dann aus Bedingung III.6 und der Tatsache, dass Wörterbucheinträge, zum Beispiel in SMILES Arbitrary Target Specification (SMARTS) Notation, atomar sind. Das Gesamtgewicht G ist dann identisch mit der *Anzahl* möglicher Elemente im Wörterbuch. Diese ist dann ebenfalls positiv ganzzahlig.

III.3.2 Vollständige Enumeration

Eine vollständige Suche nach dem optimalen reduzierten Wörterbuch muss alle möglichen reduzierten Wörterbücher überprüfen, die sich durch Kombination $\binom{n}{k}$ aus n Elementen bilden lassen.

Es handelt sich also kombinatorisch um „*Ziehen ohne Zurücklegen ohne Berücksichtigung der Reihenfolge*“. Die Anzahl möglicher reduzierter Wörterbücher beträgt dann:

$$\frac{n!}{(n-k)! \cdot k!}$$

Für beispielsweise $k = 64$ und $n = 225$ gibt es somit bereits

$$\frac{225!}{(225-64)! \times 64!} = 1,307\,528\,535\,577\,940\,956\,327\,048\,946\,925\,7 \times 10^{57}$$

mögliche reduzierte Wörterbücher.

Diese Methode ist also nur für sehr kleine Lösungsräume geeignet. Sie findet aber sicher ein globales Optimum, sofern eines existiert, da *jede* mögliche Lösung überprüft wird.

III.3.3 Stochastische Optimierung

Hypothese 1. *Der Gesamtnutzen U des reduzierten Wörterbuchs ist* ausschließlich *eine Funktion der Kombination seiner Elemente $\vec{e}_j$ und kann daher nicht a priori aus den Einzelnutzen der Elemente des Basiswörterbuchs ermittelt werden.*

$$U = F'(\vec{e}_j)$$

Unter Hypothese 1 liefern nur solche Methoden optimierte Lösungen, die nicht versuchen den Gesamtnutzen U als Funktion der Einzelnutzen von $\vec{e}_j$) zu optimieren. Solche Methoden liefert die stochastische Optimierung.

Genetischer Algorithmus

Die von Holland (vgl. [Hol92]) entwickelten genetischen Algorithmen gehören zur Klasse der evolutionären Algorithmen, welche evolutionäre Prozesse aus der Natur als Strategie zur Lösung von Optimierungsproblemen adaptieren. Dabei bilden genetische Algorithmen den biologischen Evolutionsprozess, bestehend aus Selektion, Kreuzung und Mutation auf eine Population virtueller Organismen ab. Abbildung III.5 zeigt den schematischen Ablauf eines genetischen Algorithmus.

Genetischer Algorithmus

	Erzeuge Startpopulation	
Abbruchkriterium noch nicht erreicht		
	Bewerte alle Chromosomen: $\phi(\vec{e}_j) = F'(\Gamma(\vec{e}_j))$	
	Auswahl der fittesten Chromosomen	
	Reproduktion der Chromosomen	
	Auswahl von Chromosomenpaaren für die Kreuzung	
	Kreuze Chromosomen	
	Mutiere zufällig ausgewählte Gene	
	Gebe fittestes Chromosom zurück	

Abbildung III.5: Schematischer Ablauf eines genetischen Algorithmus
Quelle: Eigene Darstellung

Die virtuellen Organismen werden durch ihre Chromosomen repräsentiert, die wiederum aus $1 \ldots n$ Genen bestehen. Ein Chromosom kodiert jeweils eine möglichen Lösung des Problems, jedes Gen kodiert dann einen Teil der Lösung. Ein mögliches Gen für das Problem DICTIONARY ist zum Beispiel ein String, der ein SMARTS kodiertes Substruktumuster enthält.

Die Bewertung der Fitness der einzelnen Individuen einer Population erfolgt mit Hilfe einer Fitnessfunktion, wie sie in Definition 6 bereits beschrieben wurde. Diese Fitnessfunktion enthält kein zusätzliches Wissen, zum Beispiel über die Struktur des Lösungsraumes.

Entscheidungen über die Auswahl der Partner für eine Kreuzung oder die Mutation eines Gens enthalten Zufallselemente, um es dem Algorithmus zu ermöglichen, auch Lösungen zu evaluieren, die nicht bereits in der Ursprungspopulation enthalten waren.

Genetische Algorithmen zeigen also folgende Charakteristika (vgl. [Gol89, S. 7]):

1. Sie arbeiten mit einer Kodierung der Parametermenge, nicht der Parameter selber
2. Sie arbeiten mit Lösungsmengen
3. Sie benutzen eine problemadäquate Fitnessfunktion zur Bewertung der Lösungen, haben aber darüber hinaus kein zusätzliches Wissen über das Problem
4. Sie enthalten nichtdeterministische Elemente, um ihre Suche nach Optima im Lösungsraum zu verbessern

Die Gene des Chromosoms mit der absolut besten Fitness aller Generationen bestimmen dann den Inhalt des optimierten reduzierten Wörterbuchs.

Genetische Algorithmen liefern typischerweise auch bei Problemen mit algorithmisch schlechter Laufzeit gute Ergebnisse. Das *Schema-Theorem* und die *Generative-Fixation-Theorie* versuchen, dieses Phänomen zu erklären.

Als *Schema* wird dabei ein Ähnlichkeitsmuster bezeichnet, welches eine Untermenge der möglichen Gene beschreibt, die an bestimmten Positionen Ähnlichkeiten aufweisen (vgl. [Gol89, S. 19]).

Schemata, welche einen überproportional hohen Beitrag zur Fitness eines Chromosoms beitragen und aufgrund ihrer Kürze wenig anfällig gegenüber Zerstörung durch Mutation oder Kreuzung sind, werden als *building blocks* bezeichnet (vgl. [Gol89, S. 20]).

Das Schema-Theorem postuliert nun, dass die building blocks von Generation zu Generation exponentiell steigende Chancen zur Reproduktion bekommen (vgl. [Gol89, S. 20] und [Bäc96, S. 126]). Da jedes Chromosom eine Vielzahl an Schemata enthält, werden die Schemata *implizit parallel* bewertet. Die obere Grenze für die erreichbare Parallelität ist $\mathcal{O}(n^3)$ für n Gene (vgl. [Gol89, S. 40f.]).

Die Beweisführung für das Schema-Theorem gilt allerdings nur unter bestimmten Randbedingungen, zum Beispiel müssen die Gene binär kodiert sein, und erklärt nicht, warum genetische Algorithmen in der Praxis auch unter anderen Bedingungen gute Ergebnisse liefern (vgl. [Bäc96, S. 128]).

Eine alternative Erklärung liefert die 2009 publizierte Generative-Fixation-Theorie (vgl. [Bur09]), über deren Güte zum jetzigen Zeitpunkt aber noch keine Aussage gemacht werden kann.

Letztendlich ist die Frage, *warum* genetische Algorithmen vergleichsweise gut funktionieren, zur Zeit als noch ungeklärt zu betrachten.

Sampling

Der Sampling Algorithmus bildet solange durch zufällige Kombination von Mustern aus dem Basiswörterbuch neue reduzierte Wörterbücher, bis ein Abbruchkriterium ihn beendet, er wählt also rein zufällig Lösungen (Samples) aus dem Lösungsraum.

Sampling Algorithmus

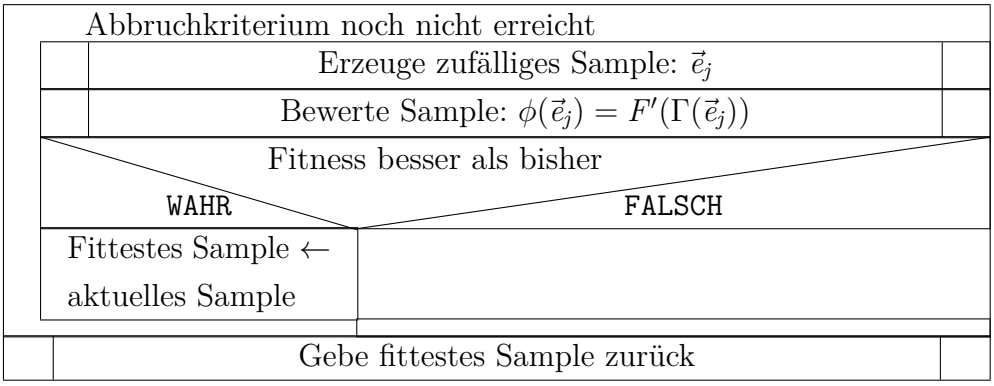


Abbildung III.6: Schematischer Ablauf des Sampling Algorithmus
Quelle: Eigene Darstellung

Jedes Sample wird, analog zum genetischen Alogrithmus, mittels der Fitnessfunktion auf seinen Selektivitätsgewinn hin bewertet. Das Sample mit dem höchsten Selektivitätsgewinn wird nach Beendigung des Algorithmus zurückgegeben. Abbildung III.6 zeigt den schematischen Ablauf des Sampling Algorithmus.

III.3.4 Lineare Optimierung

Unter der Hypothese 2 und durch Ersetzen der ursprünglichen Ganzzahligkeitsbedingung III.5 durch die relaxierte Bedingung III.8 kann DICTIONARY als Lineares Programm (LP), wie in Abbildung III.7 gezeigt, formuliert werden [MT90, S. 16].

Hypothese 2. *Der Einzelnutzen jedes Musters im Basiswörterbuch kann a priori bestimmt werden. Der Gesamtnutzen U des reduzierten Wörterbuchs ist eine Funktion der Einzelnutzen $u_j = F'(e_j)$ seiner Elemente e_j.*

$$U = \sum_{j=1}^{n} u_j = \sum_{j=1}^{n} F'(e_j)$$

Für das relaxierte LP aus Abbildung III.7 gibt es *genau ein* Element, welches nicht mehr oder nur noch anteilig in den Rucksack passt: „...the fact that solution $\overline{x}$ of the continuous relaxation of the problem has only one fractional variable...“ [MT90, S. 27]. Dieses wird als das *kritische Element* bezeichnet: „...until the first item, s, is found which does not fit. We call it the *critical item*...“ [MT90, S. 16].

$$\begin{aligned} U \rightarrow Max = f(x) &= u_1 \cdot x_1 + \cdots + u_j \cdot x_j \\ x_1 + \cdots + x_j &\leq G \\ u_j &\geq 0 \\ \\ 0 \leq x_j &\leq 1 \end{aligned} \tag{III.8}$$

Abbildung III.7: Allgemeines lineares Programm
Quelle: Eigene Darstellung (vgl. [MM92, S. 88ff.] oder [DK92]

Sind die Nutzen absteigend sortiert $u_1 \geq u_2 \geq \cdots \geq u_n$, ist das *kritische Element* immer dasjenige Element mit dem niedrigsten Nutzen, welches nicht mehr oder nur noch anteilig dem Rucksack hinzugefügt werden kann: $0 \leq \overline{x}_j \leq 1$. (vgl. [MT90, S. 27ff.])

Eine näherungsweise optimale Lösung für DICTIONARY in Form eines LP kann dann mit einem geeigneten Algorithmus gefunden werden, wenn $\overline{x}_j = 0$ gesetzt wird: „Setting this variable to 0 gives a feasible solution to KP. . .We can expect that z' [die näherungsweise Lösung; Anm. d. Verf.] is, on average, quite close to the optimal solution value z." [MT90, S. 27-28]

Um DICTIONARY unter Hypothese 2 formulieren zu können, muss außerdem der Einzelnutzen eines Musters u_j bestimmt werden können. Da hier jedes Muster isoliert betrachtet werden muss, wurde ein statistischer Lösungsansatz auf Basis von Hypothese 3 gewählt.

Hypothese 3. *Je mehr Bits tatsächlich genutzt werden, desto mehr unterschiedliche Fingerprints sind theoretisch möglich. Bei optimaler Ausnutzung der* 64×8 *Bits eines FPPC8 Fingerprints (vgl. Anhang D) liegt die Obergrenze bei* 2^{512} *möglichen Fingerprints.*

Eine untere Grenze kann unter der Annahme, dass jedes Muster für sich betrachtet, wie in Tabelle D.1 gezeigt, 9 Zustände kodiert, dann analog als 9^{64} *mögliche Fingerprints bestimmt werden.*

$$9^{64} \leq S \leq 2^{512} \tag{III.9}$$

Werden nicht alle Bits ausgenutzt, sinkt die theoretisch mögliche Selektivität: Im Falle eines nicht benutzen Bits zum Beispiel auf $2^{511}/2^{512} = 1/2 = 50\%$ *der theoretischen Obergrenze und auf* $8 \times 9^{63}/9^{64} = 8/9 \approx 89\%$ *der theoretischen Untergrenze.*

Je näher also die Realisationen (als Bitmuster) eines Musters an der diskreten Gleichverteilung liegen, desto mehr Bits werden tatsächlich genutzt und desto höher ist die theoretische Selektivität des Fingerprints. u_j *kann folglich mittels einer Maßzahl für die Anpassung der Bitmuster an eine diskrete Gleichverteilung bestimmt werden.*

Eine solche Maßzahl ist der χ^2 Wert, wie er für einen χ^2-Anpassungstest nach PEARSON benutzt wird.

1. Für jedes Muster j im Basiswörterbuch wird zunächst die Anzahl der verschiedenen Bitmuster h_i bestimmt, die es auf T_Q angewendet generiert
2. Für jedes Muster j im Basiswörterbuch wird sein χ_j^2 Wert (Gleichung III.12) berechnet
3. Der χ_j^2 Wert wird als Maßzahl für die Annäherung an eine diskrete Gleichverteilung der Bitmuster benutzt

Betrachtet man jedes mögliche Bitmuster als Realisation eines Wurfs eines perfekten Würfels mit 9 Seiten, entspricht die Wahrscheinlichkeit für das Auftreten jedes Bitmusters der Wahrscheinlichkeitsfunktion $P(x)$ der diskreten Gleichverteilung (III.10).

$$\mathrm{P}(x) = \begin{cases} \frac{1}{n} & \text{für } x = x_i (i = 0, \ldots, 8) \\ 0 & \text{sonst} \end{cases} \tag{III.10}$$

Jedes Bitmuster tritt also mit einer Wahrscheinlichkeit von $1/9$ auf. Der Erwartungswert $E(h)$ ist dann das Produkt aus der Anzahl der Tupel in T_Q und $1/9$ (III.11).

$$E(h) = \frac{1}{9} \times \text{Anzahl der Tupel in } T_Q \tag{III.11}$$

$$\chi_j^2 = \sum_{i=0}^{8} \frac{(h_{i,j} - E(h))^2}{E(h)} \tag{III.12}$$

$$u_j = \chi_j^2$$

Der χ^2 Wert kann also gemäß III.12 berechnet werden.

Da χ^2 mit zunehmender Anpassung der beobachteten Werte an die Erwartungswerte der diskreten Gleichverteilung *sinkt*, muss entweder das LP aus Abbildung III.7 in die Form $Min.U = f(x) = u_1 \cdot x_1 + u_2 \cdot x_2 + u_3 \cdot x_3 + \cdots + u_j \cdot x_j$ umgewandelt werden oder man bestimmt u_j als $1/\chi_j^2$ wobei der dann mögliche Fall der Division durch Null gesondert berücksichtigt werden muss.

Simplex-Algorithmus

Der Simplex-Algorithmus wurde 1951 erstmals von Dantzig (vgl. [Dan51]) publiziert und ist ein Standardverfahren zur Lösung linearer Programme. Eine formale Beschrei-

bung des Simplex-Algorithmus findet sich zum Beispiel in [KV05, S. 53-56] oder [DK92, S. 23-73].

Eine obere Grenze für die Laufzeit des Simplex-Algorithmus ist $\mathcal{O}(2^n)$ für n Variablen und $2n$ Nebenbedingungen (vgl. [KM72]). Allerdings ist die durchschnittliche Laufzeit für zufällige Modelle polynomial (vgl. [Bor82]).

Eine exemplarische Formulierung eines relaxierten binären Rucksackproblems der Form $Min.U = f(x) = u_1 \cdot x_1 + u_2 \cdot x_2 + u_3 \cdot x_3 + \cdots + u_j \cdot x_j$ für den Simplex-Solver des Computer Algebra Systems MAXIMA (vgl. [Max09]) ist, zusammen mit der gefundenen Lösung, in Anhang G aufgeführt.

Greedy-Algorithmus

Bei genauerer Betrachtung der Lösung des LP in Anhang G zeigt sich, dass die Lösung die Basisvariablen in absteigender Sortierung der Nutzen $u_1 \geq u_2 \geq \cdots \geq u_n$ enthält. Werden die Elemente x_j absteigend nach ihrem Nutzen vorsortiert

$$u_1 \geq u_2 \geq \cdots \geq u_n \tag{III.13}$$

verhält sich der Simplex-Algorithmus äquivalent zu einem Greedy-Algorithmus der in Abbildung III.8 gezeigten Form.

Denn setzt man für das kritische Element $\overline{x}_j = 0$, dann kann das binäre Rucksackproblem auch mit einem Greedy-Algorithmus gelöst werden: „If the items are sorted as in (2.7) [Absteigend nach Nutzen; Anm. d. Verf.], a more effective algorithm is to consider them [die items; Anm. d. Verf.] according to increasing indices and insert each new item into the knapsack if it fits.“ [MT90, S. 28]

Greedy-Algorithmus

Bestimme den individuellen Nutzen jedes Elements
Sortiere absteigend nach individuellem Nutzen
Wähle die ersten G Elemente aus

Abbildung III.8: Schematischer Ablauf des Greedy-Algorithmus
Quelle: Eigene Darstellung

$\overline{x}_j = 0$ ergibt sich für DICTIONARY aus der Annahme aller Gewichte $g_j = 1$ in Verbindung mit der Bedingung III.7. Damit gilt in diesem Fall $z = z'$.

Die ersten G Elemente bestimmen dann den Inhalt des optimierten reduzierten Wörterbuchs.

Die obere Grenze der Laufzeit des Greedy-Algorithmus ist $\mathcal{O}(n \log n) + \mathcal{O}(g)$ für die Sortierung von n Elementen mit irgendeinem effizienten Sortieralgorithmus und anschließender linearer Auswahl der ersten g Elemente: „The time required to sort N records, using a decent general-purpose sorting algorithm, is roughly proportional to $N \log N$; we make about $\log N$ "passes" over the data. This is the minimal possible time. . .“ [Knu98, S. 5].

IV Experimentelle Untersuchung ausgewählter Verfahren zur dynamischen Optimierung der Indexselektivität chemischer Datentypen in relationalen Datenbankmanagementsystemen

On two occasions I have been asked [by members of Parliament], "Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?" I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question. - *C. Babbage*

IV.1 Versuchsumgebung

Mit der in Abschnitt 5.5 beschriebenen Aussnahme wurden alle Experimente auf einem Notebook mit Intel® Core™ Duo T2450 Mikroprozessor mit 2 GHz Taktfrequenz und 2 GB Hauptspeicher unter dem Betriebssystem Windows® XP Professional (32 Bit, Service Pack 3) durchgeführt. Entwickelt wurde mit Eclipse 3.2.0 (vgl. [Ecl09]), Java™ Development Kit (JDK) 5 (vgl. [Mic]) und MinGW 5.1.3 (gcc 3.4.5) (vgl. [tea09]). Der verwendete genetische Optimierer wurde mit JGAP (vgl. [MMV+09]) in der Programmiersprache Java, alle Datenbankprozeduren wurden in pl/pgSQL implementiert.

Für alle Algorithmen, die Zufallszahlen benötigen, wurde, falls nicht anders angegeben, der Mersenne Twister Algorithmus „MT19937: Integer version“ verwendet, ein Pseudozufallszahlengenerator mit einer Periode von $2^{19937} - 1$ (vgl. [MN98]).

Als Werkzeug für die Versuchsplanung wurde Design-Ease® (vgl. [DE707]) in der Version 7.2.1 P verwendet.

IV.2 Versuchsdaten

Als Datenquellen für die Experimente dienten die bereits in Abschnitt III.2, Tabelle III.2 gezeigten Chemikalienkataloge. Der ChemCollect Katalog wurde allerdings aufgrund seiner geringen Größe nicht verwendet und durch den Datenbestand eines Laborlogistiksystems der BBS ersetzt.

Tabelle IV.1: Für die Experimente verwendete Chemikalienkataloge
Quelle: Eigene Darstellung

Katalog	Jahr	Strukturen	Kurzbezeichnung
Maybridge Screening Compounds	2008	58 159	MAYSC
Asinex Platinum Collection	2008	125 231	ASINEX_PC_2008
Asinex Gold Collection	2008	229 398	ASINEX_GC_2008
BBS VC	2009	1 700 000	BBS

Tabelle IV.1 zeigt die daraus resultierende Liste der Datenquellen zusammen mit ihren in der Versuchsdokumentation verwendeten Kurzbezeichnungen.

Als Basiswörterbuch wurde die in Anhang J gezeigte modifizierte Version der veröffentlichten Definition des PubChem Fingerprints [Pub09a] verwendet. Die Modifikationen bestanden darin die Datei in ein OpenBabel kompatibles Format zu transformieren und einige Muster zu entfernen, die zu dem in Anhang E beschriebenen overtraining führten.

IV.3 Versuchsplanung

IV.3.1 Versuchsplanung zur Ermittlung der korrekten Parametrierung der Optimierungsalgorithmen

Für die Planung und Auswertung der einzelnen Versuche zur Ermittlung der korrekten freien Parameter für die Optimierungsalgorithmen, auch als *Behandlungsfaktoren* im Sinne der Definition 7 bezeichnet, wurde bei einem Behandlungsfaktor die univariate Vorgehensweise, auch als One-Factor-at-a-Time (OFAT) oder *ceteris paribus* Methode bekannt, verwendet.

Definition 7. *„Ist ein Faktor während der Dauer eines Versuches fest, d.h. hat er nur eine Stufe, so bezeichnen wir ihn als* Konstantfaktor*. Er ist dann Bestandteil der Versuchsbedingungen. Besteht das Ziel eines Versuches darin, den Einfluss eines Faktors zu untersuchen, indem man seine Stufen systematisch variiert, so nennen wir einen solchen Faktor einen* Planfaktor*. Ist die Ermittlung der möglichen Effekte eines Planfaktors Teil der (oder die) Versuchsfrage, so heißt ein solcher Planfaktor* Behandlungsfaktor*. Die übrigen Planfaktoren heißen* Blockfaktoren*. Alle übrigen Faktoren, die man nicht systematisch variieren kann, die aber Einfluss auf die Versuchsergebnisse haben, heißen* Restfaktoren*. Die Rest- und die Blockfaktoren zusammen bilden die* Störfaktoren*“* [RVG07, S. 14]

Bei mehr als einem Behandlungsfaktor wurde die Methode des Design of Experiments (DOE) mit 2^k voll-faktoriellen Plänen verwendet, um die Anzahl der notwendigen Behandlungen im Sinne der Definition 8 zu minimieren. Ergebnis jeder Planung ist ein randomisierter *Versuchsplan* mit typischerweise mehreren Behandlungen.

Definition 8. *„Wird in einem Versuch nur ein Behandlungsfaktor untersucht, so heißen seine Stufen* Behandlungen*. Im Falle mehrerer Behandlungsfaktoren bezeichnet man die in den Versuch einzubeziehenden* Stufenkombinationen *der Behandlungsfaktoren als Behandlungen.“* [RVG07, S. 16]

Jedem Versuchsplan wurde ein zusätzlicher Zentralpunkt hinzugefügt, der genau auf den arithmetischen Mitteln aller Behandlungsfaktoren liegt. Er dient dazu, eine eventuell vorhandene Krümmung des Ergebnisgraphen zu entdecken. In den abgebildeten Versuchsplänen ist der Zentralpunkt jeweils durch einen grauen Hintergrund der entsprechenden Zeile markiert.

Der Experiment Runner

Die mit Design-Ease® erstellten Versuchspläne können im EXtensible Markup Language (XML) Format exportiert werden. Es lag daher nahe, die Versuche automatisch auf Basis dieser XML Versuchspläne durchführen zu lassen. Zu diesem Zweck wurde das Programm „Experiment Runner“ in Java entwickelt. Abbildung IV.2 zeigt das Struktogramm des „Experiment Runner“ zusammen mit seinem zentralen Unterprogramm „Behandlung durchführen“.

```
<experiment name="" description="">
<algorithm>GA</algorithm>
<designfile>
c:/tigress/experiments/ex3.xml
</designfile>
<datatable>
maysc
</datatable>
</experiment>
```

Abbildung IV.1: Aufbau des „Experiment Control File“
Quelle: Eigene Darstellung

Die Ergebnisse jeder Behandlung wurden nicht direkt im Versuchsplan sondern in getrennten Textdateien protokolliert. Dies war nötig, da keine Metadaten, zum Beispiel das jeweilig erzeugte reduzierte Wörterbuch, mit in der XML Datei gespeichert werden können.

Neben dem eigentlichen Versuchsplan benötigt der „Experiment Runner“ noch eine, in Abbildung IV.1 gezeigte XML Steuerdatei (das „Experiment Control File“), in welcher der zu verwendende Algorithmus, der Versuchsplan und die zu optimierende Basistabelle spezifiziert werden.

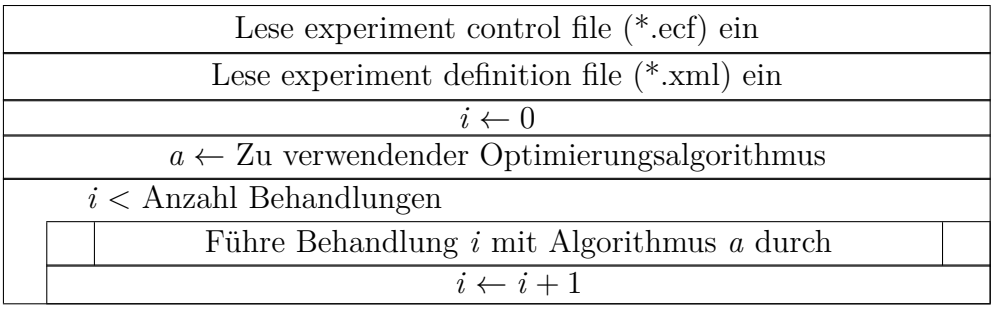
Der Aufruf *java de.furaffinity.diss.ExperimentRunner <Experiment Control File>* startet dann die automatische Abarbeitung des gewünschten Versuchsplans. Das selektivste gefundene reduzierte Wörterbuch wird als Ergebnis des Versuchsplans ausgegeben.

Die so gewonnenen reduzierten Wörterbücher werden dabei wie folgt bezeichnet:

<Algorithmus>_(optional mit Stichprobe)_<Anzahl Einträge im reduzierten Wörterbuch>_<Code der Basistabelle> (analog zu Tabelle IV.1).

Die Bezeichnung G_S_64_MAYSC bedeutet also zum Beispiel ein durch Genetischer Algorithmus (GA) mit Stichprobe auf der Tabelle MAYSC erzeugtes reduziertes Wörterbuch mit 64 Einträgen.

Experiment Runner



Behandlung durchführen

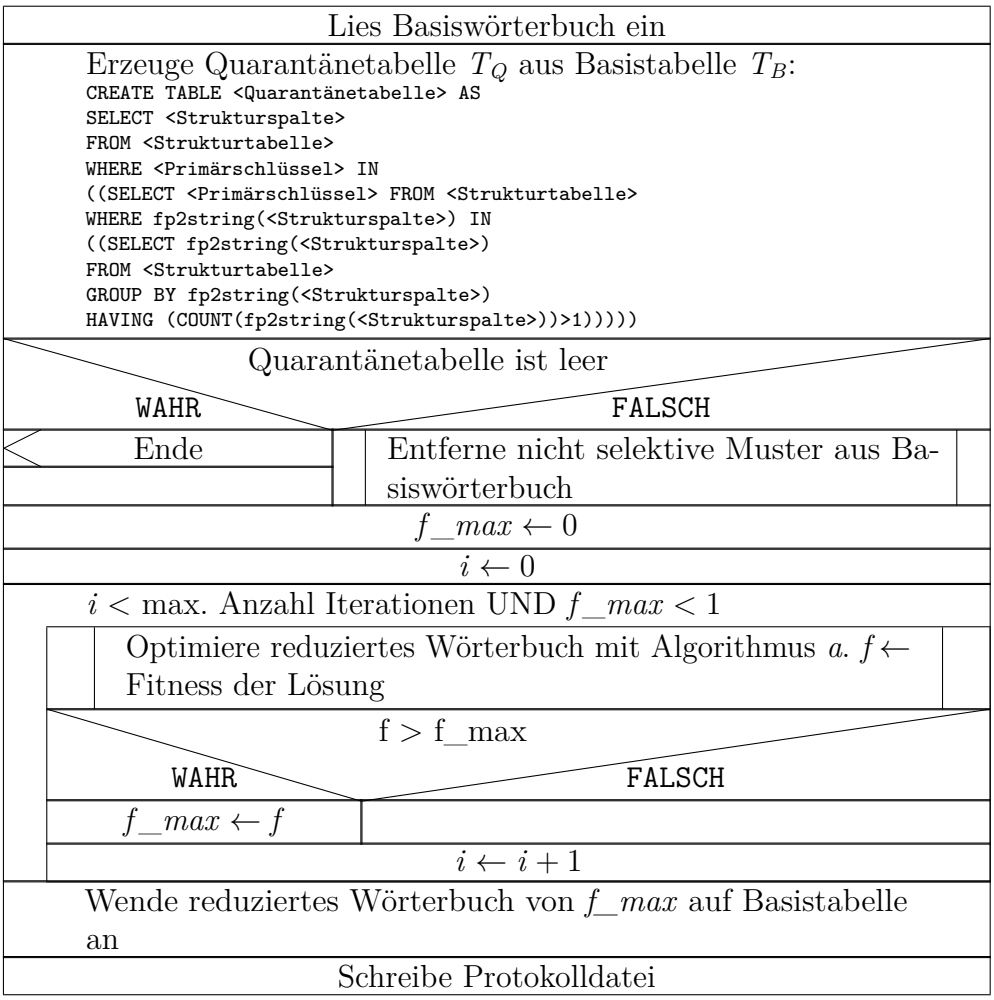


Abbildung IV.2: Schematischer Ablauf des „Experiment Runner“ mit Unterprogramm „Behandlung durchführen“
Quelle: Eigene Darstellung

Lineare Optimierung Da nur ein Behandlungsfaktor beeinflusst werden kann, wurde hier die OFAT Methode gewählt und der Behandlungsfaktor Wörterbuchgröße wie im in Tabelle IV.2 gezeigten Intervall variiert. Daraus resultiert der in Tabelle IV.3 gezeigte Versuchsplan.

Tabelle IV.2: Behandlungsfaktoren und Beobachtungen des LP Algorithmus
Quelle: Eigene Darstellung

Behandlungsfaktoren	
Faktor	**Faktorstufen**
Wörterbuchgröße	8,16,24,32,40,48,56,64
Konstantfaktoren	
Faktor	**Faktorstufen**
Tabelle	MAYSC
Beobachtungen	
Δ_S	
t_{Run}	

Tabelle IV.3: Versuchsplan des LP Algorithmus
Quelle: Eigene Darstellung

		Factor 1	Response 1	Response 2
Std	Run	A:Wörterbuchgröße Gene	Δ_S	t_{Run} msec
1	1	8		
2	2	16		
3	3	24		
4	4	32		
5	5	40		
6	6	48		
7	7	56		
8	8	64		

Sampling Die in Tabelle IV.4 aufgeführten Behandlungsfaktoren führen zu dem in Tabelle IV.5 gezeigten mehrfaktoriellen Versuchsplan.

Tabelle IV.4: Behandlungsfaktoren und Beobachtungen des Sampling Algorithmus
Quelle: Eigene Darstellung

Behandlungsfaktoren	
Faktor	**Faktorstufen**
Samplegröße	8,64
Anzahl Samples	1,100
Konstantfaktoren	
Faktor	**Faktorstufen**
Tabelle	MAYSC
Beobachtungen	
Δ_S	
t_{Run}	

Tabelle IV.5: 2^2 Versuchsplan des Sampling Algorithmus
Quelle: Eigene Darstellung

		Factor 1	Factor 2	Response 1	Response 2
Std	Run	A:Evolutions Chromosome	B:Chromosome Size Gene	Δ_S	t_{Run} msec
4	1	64	100		
1	2	8	1		
2	3	64	1		
5	4	36	51		
3	5	8	100		

Genetische Optimierung Die in Tabelle IV.6 aufgeführten Behandlungsfaktoren führen zu dem in Tabelle IV.7 gezeigten mehrfaktoriellen Versuchsplan.

Tabelle IV.6: Behandlungsfaktoren und Beobachtungen des genetischen Algorithmus
Quelle: Eigene Darstellung

Behandlungsfaktoren	
Faktor	**Faktorstufen**
Chromosome Size	8,64
Population Size	1,100
Evolutions	1,100
Konstantfaktoren	
Faktor	**Faktorstufen**
Tabelle	MAYSC
Beobachtungen	
Δ_S	
t_{Run}	

Tabelle IV.7: 2^3 Versuchsplan des genetischen Algorithmus
Quelle: Eigene Darstellung

		Factor 1	Factor 2	Factor 3	Response 1	Response 2
Std	Run	A:Population Size Chromosome	B:Evolutions	C:Chromosome Size Gene	Δ_S	t_{Run} msec
2	1	100	1	8		
4	2	100	100	8		
7	3	1	100	64		
9	4	51	51	36		
5	5	1	1	64		
8	6	100	100	64		
6	7	100	1	64		
1	8	1	1	8		
3	9	1	100	8		

IV.3.2 Versuchsplanung zur Ermittlung der Auswirkungen des optimierten Index auf die Suchperformance

Um festzustellen, wie sich die Optimierung der Fitnessfunktion ϕ auf die tatsächliche Performance des optimierten Fingerprintindex auswirkt, wurden standardisierte Testsuchen auf der Basistabelle ausgeführt. Dies wurde jeweils mit unoptimiertem Fingerprintindex und mit optimiertem Fingerprintindex gemacht.

```
CREATE OR REPLACE FUNCTION catalogdata.maysc_timings()
  RETURNS void AS
$BODY$
DECLARE start_time timestamp;
DECLARE curr_row record;
BEGIN

FOR curr_row IN SELECT id FROM catalogdata.testsearches_infochem_standardized LOOP

  start_time=timeofday()::timestamp;

  INSERT INTO catalogdata.timings_baseline (SELECT curr_row.id, count(1),
  timeofday()::timestamp-start_time FROM catalogdata.maysc WHERE
  structure >= (SELECT structure FROM catalogdata.testsearches_infochem_standardized
  WHERE id=curr_row.id));

END LOOP;
END;
$BODY$
  LANGUAGE 'plpgsql' VOLATILE;
```

Listing IV.1: Datenbankprozedur zur Messung der Suchperformance für $T_B = MAYSC$
Quelle: Eigene Darstellung

Das verwendete Testset wurde aus einem Testset abgeleitet, welches die Firma Infochem GmbH zur Validierung ihrer kommerziellen Cartridge benutzt und freundlicherweise für diese Arbeit zur Verfügung gestellt hat [Kra08]. Es wurden folgende Modifikationen vorgenommen:

- Drei Strukturen wurden entfernt, weil sie von PGCHEM::TIGRESS als fehlerhaft abgelehnt wurden, da ein als „H+“ definiertes Atom vom Parser nicht verarbeitet werden kann

- Strukturen, die aus mehreren unverbundenen Teilen bestehen, wurden entfernt, da PGCHEM::TIGRESS solche Suchen nicht unterstützt
- Das Testset wurde um Strukturen bereinigt, die PGCHEM::TIGRESS als Duplikate betrachtet, so dass diese nur noch einmal vorkommen

Von 20 000 Strukturen des originalen Testsets bleiben nach oben genannten Modifikationen 16 587 Strukturen übrig, die für die Versuche verwendbar sind.

Die Messung selbst wurde mittels der in Listing IV.1 (exemplarisch für die Basistabelle MAYSC) gezeigten Datenbankprozedur vorgenommen, welche für alle Zeilen im Testset eine Substruktursuche auf der Basistabelle durchführt und die Ergebnisse in einer weiteren Datenbanktabelle protokolliert. Aufgezeichnet wurden dabei der Primärschlüssel der gesuchten Substruktur, die Anzahl der Treffer sowie die Dauer der Suche. Die Ergebnisse werden in Abschnitt IV.4 gezeigt.

IV.3.3 Versuchsplanung zur Abschätzung der Änderungsstabilität des optimierten Index

Um die Frage zu beantworten, wann eine erneute Optimierung von T_B notwendig wird, wenn sich ihr Inhalt zum Beispiel durch INSERT und UPDATE Operationen gegenüber dem Zeitpunkt der letzten Optimierung verändert hat, wurden an eine T_B mit optimiertem Index neue Daten per INSERT aus einer anderen Tabelle angehängt und danach das resultierende Δ_S gemessen.

Aus dieser Überlegung ergeben sich die in Tabelle IV.8 gezeigten Behandlungsfaktoren für diesen Versuch.

Als Ausgangspunkt dient die Tabelle MAYSC, die Daten für den INSERT stammen aus ASINEX_PC_2008 und ASINEX_GC_2008.

MAYSC dient hier als erstes Beispiel für einen spezifischen Datenbestand. Es handelt sich um einen Katalog der speziell für Schritt 3 des Drug Discovery Process zusammengestellt wurde und überwiegend inhouse synthetisierte Strukturen des Anbieters enthält:

„The Maybridge Screening collection consists of over 56,000 organic compounds, largely produced by us at Maybridge. These are individually designed compounds, produced

Tabelle IV.8: Behandlungsfaktoren und Beobachtungen des Änderungsstabilitätsexperiments
Quelle: Eigene Darstellung

Behandlungsfaktoren	
Faktor	**Faktorstufen**
Größe T_B%	100,200,400
Wörterbuch	L_64_MAYSC, G_64_MAYSC
Konstantfaktoren	
Faktor	**Faktorstufen**
T_B	MAYSC
Beobachtungen	
Δ_S	

by innovative synthetic techniques, based on over 45 years of experience in heterocyclic chemistry.“ [May08b]

ASINEX_PC_2008 dient als zweites Beispiel für einen spezifischen Datenbestand, da er laut Anbieter nur inhouse synthetisierte Strukturen enthält und speziell für das Screening im Drug-Discovery Prozess zusmmengestellt wurde:

„This collection is generally more lead-like than the Gold Collection and as it is an in-house collection we are able to provide more efficient follow-up services. The Platinum Collection consists of 150,000 compounds which are only available from ASINEX.“ [Asi08c]

ASINEX_GC_2008 ist ein Beispiel für einen unspezifischen Datenbestand, da ihn der Anbieter aus bestehenden Katalogen aus dem akademischen Sektor der ehemaligen Sowjetunion zusammengestellt hat und eine hohe chemische Diversität bescheinigt:

„The Gold Collection started in 1994 with the collecting of compounds from former Soviet universities and research institutes and supplying them on to Western Pharma and Biotech companies. It is an extremely diverse library which we estimate covers 85 % of the chemical space of other historical libraries. The Gold Collection consists of 250,000 compounds and its greatest strength is the level of diversity.“ [Asi08c]

Diese Auswahl der Testdaten ist natürlich teilweise willkürlich, aber da es nur um eine Abschätzung der Änderungsstabilität geht und außerdem keine Annahmen über die

vom Nutzer im konkreten Anwendungsfall tolerierte Änderung gemacht werden können, wäre die Alternative eine reine Zufallsauswahl gewesen, die keinerlei Aussagen über die Veränderung der Stabilität abhängig von der Art der Daten (spezifischer versus unspezifischer Inhalt) zulassen würde.

IV.4 Ergebnisse

Alle statistischen Berechnungen wurden mit R (vgl. [R D09]) oder DESIGN-EASE® durchgeführt. Für allgemeine Berechnungen und die Lösung der LP wurde das Computer Algebra System MAXIMA (vgl. [Max09]) eingesetzt.

Das Signifikanzniveau für die irrtümliche Ablehnung von H_0 (Fehler 1. Art) ist $\alpha = 0,05$ (5 %), falls nicht anders angegeben.

Das lineare Modell für die einfaktorielle Analyse lautet (vgl. [AW07, S.63]):

$$\hat{y} = \beta_0 + \beta_1 x \tag{IV.1}$$

Das lineare Modell für die mehrfaktorielle Analyse lautet (vgl. [AW07, S.63]):

$$\hat{y} = \beta_0 + \beta_i x_j + \beta_i x_j + \beta_{ij} x_i x_j, \quad i = 1, \ldots, k,\ j = 1, \ldots, l \tag{IV.2}$$

Für jedes Ergebnis wurde überprüft, ob die Art und Anzahl der Treffer der unoptimierten Messung mit der der optimierten Messung übereinstimmen, um gegebenenfalls Overtraining zu entdecken. Dazu wurde das in Listing IV.2 gezeigte SQL-Statement verwendet.

```
select count(1)
from catalogdata.timings_baseline b,
catalogdata.timings_optimized o
where b.id=o.id
and b.hitcount != o.hitcount
```

Listing IV.2: SQL zur Prüfung auf Overtraining
Quelle: Eigene Darstellung

Ist das Ergebnis gleich Null, stimmen beide Messungen in Art und Anzahl der Treffer überein, und somit liegt kein Overtraining vor.

IV.4.1 Stochastische Optimierung

Sampling

Selektivität Hier wurden die in Tabelle IV.9 gezeigten Ergebnisse gemessen. Abbildung H.2 und Abbildung H.1 in Anhang H zeigen die zugehörigen Analysis of Variance (ANOVA) Protokolle für t_{Run} und Δ_S.

Zunächst konnte keine statistische Signifikanz für einen Zusammenhang zwischen den Behandlungsfaktoren und t_{Run} bzw. Δ_S festgestellt werden. Betrachtet man allerdings die response surface für t_{Run} in Abbildung IV.3, so lässt sich ein solcher Zusammenhang zwischen der Anzahl der Evolutionen und der Laufzeit zumindest vermuten.

Tabelle IV.9: 2^2 Versuchsplan des Sampling Algorithmus mit Beobachtungen
Quelle: Eigene Darstellung

		Factor 1	Factor 2	Response 1	Response 2
Std	Run	A:Evolutions Chromosome	B:Chromosome Size Gene	Δ_S	t_{Run} msec
4	1	64	100,0	0,688 2	832 292
1	2	8	1,0	0,059 3	123 952
2	3	64	1,0	0,405 6	153 687
5	4	36	50,5	0,630 4	346 248
3	5	8	100,0	0,435 3	481 966

Da die ANOVA jedoch keine Signifikanz feststellen kann, liegt hier unter Umständen ein Fehler zweiter Art vor. Aus der Kenntnis über den Ablauf des Sampling-Algorithmus lässt sich jedenfalls schließen, dass o.g. Zusammenhang tatsächlich existiert:

In Anbetracht der Tatsache, dass jede Evolution Zeit braucht, ist ein Zusammenhang der Form $t_{Run} = \text{Anzahl der Iterationen} \cdot \text{Zeit pro Iteration}$ trivial.

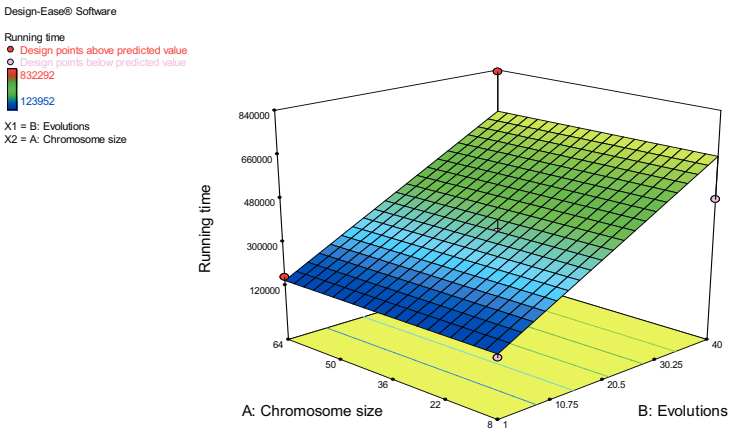


Abbildung IV.3: Response surface für t_{Run} abhängig von Chromosome Size und Evolutions
Quelle: Eigene Darstellung mit DESIGN-EASE®

Aufgrund der statistischen Unsicherheit sollte dieser postulierte Zusammenhang nur als Trend in Betracht gezogen werden.

Im Fall von Δ_S kann ein Zusammenhang jedoch ausgeschlossen werden. Da der Sampling-Algorithmus rein *zufällig* Lösungen auswählt und seine Laufzeit endlich ist, ist die Güte der Lösung weder abhängig von der Chromosomengröße noch von der Anzahl der Evolutionen. In der Stichprobe möglicher Lösungen ist die Entdeckung einer guten (oder gar der optimalen) Lösung rein zufällig.

Insofern ist die Hypothese von ANDERSON „There is now a better way. Petabytes allow us to say: "Correlation is enough." We can stop looking for models. We can analyze the data without hypotheses about what it might show. We can throw the numbers into the biggest computing clusters the world has ever seen and let statistical algorithms find patterns where science cannot.“ [And08] doch recht schwach. Auch zufällige Korrelation ist - nur zufällig.

Tabelle IV.10: Korrelationskoeffizienten für Δ_S und $\Delta_{\overline{x}}$
Quelle: Eigene Darstellung

r nach Pearsson	*rho* nach Spearman	*tau* nach Kendall
0.95	0.87	0.75

Suchperformance In der Stichprobe potentieller Lösungen ist die Entdeckung einer guten Lösung rein zufällig, daher wurde auf eine experimentelle Überprüfung der Suchperformance verzichtet.

Da Δ_S und $\Delta_{\overline{x}}$, wie in Tabelle IV.10 gezeigt, stark positiv korrelieren, kann außerdem die Annahme getroffen werden, dass das in Tabelle IV.9 erreichte beste Δ_S von 63 Prozent nur zu einem im Vergleich zu den anderen Algorithmen niedrigen $\Delta_{\overline{x}}$ führt.

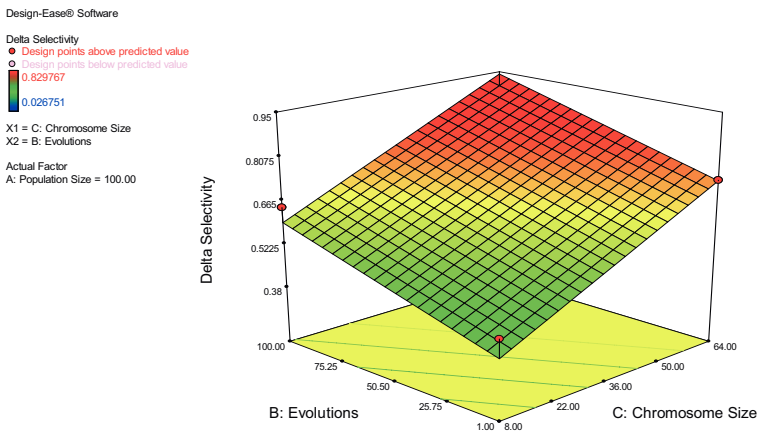
Das schließt nicht aus, dass der Sampling-Algorithmus möglicherweise bessere Lösungen finden könnte, dies konnte nur nicht beobachtet werden.

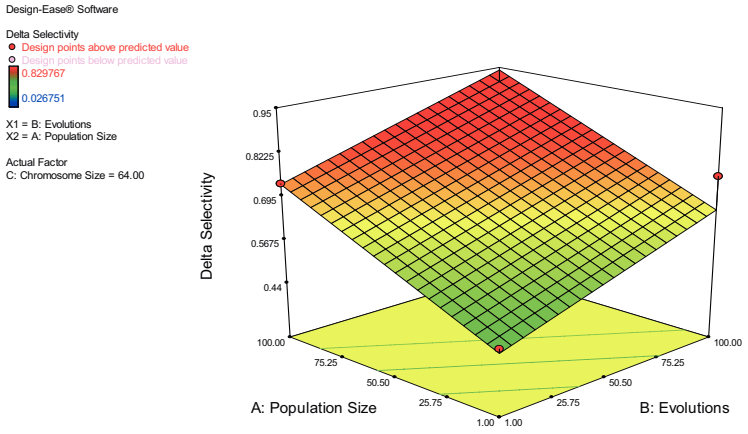
Genetische Optimierung

Selektivität Hier wurden die in Tabelle IV.11 gezeigten Ergebnisse gemessen. Abbildung H.3 und Abbildung H.4 zeigen die zugehörigen ANOVA Protokolle für Δ_S und t_{Run}. Sie zeigen eine ausreichende statistische Signifikanz für einen Zusammenhang zwischen den Behandlungsfaktoren und Δ_S beziehungsweise t_{Run}.

Tabelle IV.11: 2^3 Versuchsplan des genetischen Algorithmus mit Beobachtungen
Quelle: Eigene Darstellung

		Factor 1	Factor 2	Factor 3	Response 1	Response 2
Std	Run	A:Population Size Chromosome	B:Evolutions	C:Chromosome Size Gene	Δ_S	t_{Run} msec
2	1	100	1	8	0,446	1 488 263
4	2	100	100	8	0,645	65 158 680
7	3	1	100	64	0,756	655 781
9	4	50,5	50,5	36	0,755	20 711 679
5	5	1	1	64	0,460	20 688
8	6	100	100	64	0,830	96 785 433
6	7	100	1	64	0,734	2 044 938
1	8	1	1	8	0,027	13 453
3	9	1	100	8	0,275	466 531

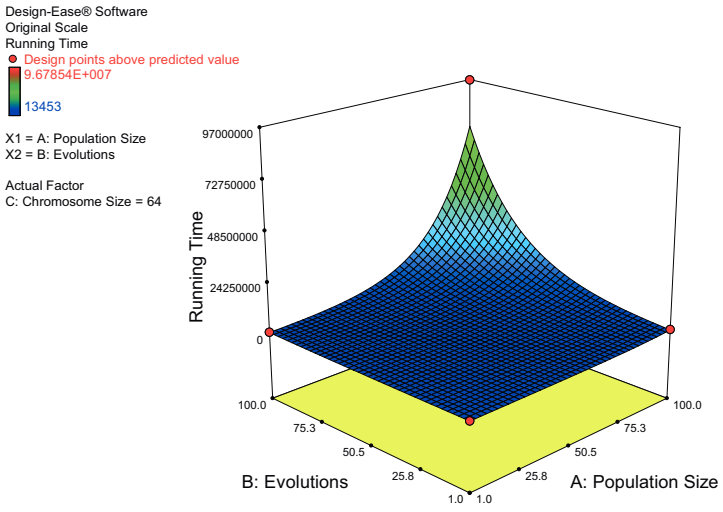




$$
\begin{aligned}
\Delta_S &= \beta_0 + \beta_1 \cdot a + \beta_2 \cdot b + \beta_3 \cdot c \\
\beta_0 &= 0,046\,773 \\
\beta_1 &= 0,002\,872\,84 \\
\beta_2 &= 0,002\,118\,71 \\
\beta_3 &= 0,006\,190\,51
\end{aligned}
$$

Abbildung IV.4: Response surfaces für Δ_S abhängig von Chromosome Size, Population Size und Evolutions mit dem zugehörigen linearen Modell
Quelle: Eigene Darstellung mit DESIGN-EASE®

Die response surfaces für Δ_S in Abhängigkeit von Chromosome Size und Evolutions sowie Δ_S in Abhängigkeit von Population Size und Evolutions in Abbildung IV.4 zeigen einen linearen Zusammenhang, während die response surface für t_{Run} in Abhängigkeit von Population Size und Evolutions in Abbildung IV.5 einen exponentiellen Zusammenhang zeigt, der es notwendig macht, t_{Run} wie in IV.4 logarithmisch zu transformieren.



$$\log 10(\Delta_S) = \beta_0 + \beta_1 \cdot a + \beta_2 \cdot b$$
$$\beta_0 = 4,150\,71$$
$$\beta_1 = 0,021\,093$$
$$\beta_2 = 0,016\,054$$

Abbildung IV.5: Response surface für t_{Run} abhängig von Population Size und Evolutions mit dem zugehörigen transformierten linearen Modell Quelle: Eigene Darstellung mit DESIGN-EASE®

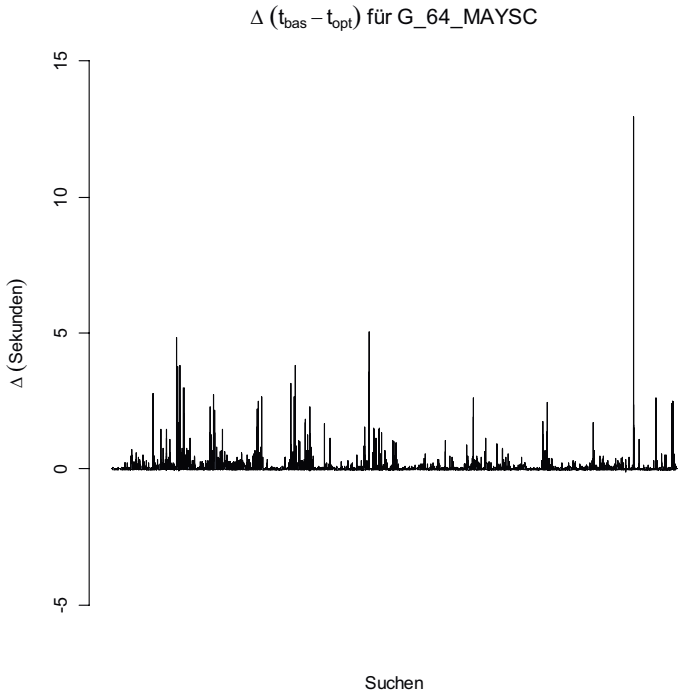
Damit ergibt sich folgendes lineares Modell zur Ermittlung der Parametrierung des GA Algorithmus:

$$\begin{aligned}\Delta_S = {} & 0,046\,773 + 0,002\,872\,84 \times \text{Population Size} \\ & + 0,002\,118\,71 \times \text{Evolutions} + 0,006\,190\,51 \times \text{Chromosome Size} \end{aligned} \quad \text{(IV.3)}$$

$$\log_{10}(t_{Run}) = 4,150\,71 + 0,021\,093 \times \text{Population Size} + 0,016\,054 \times \text{Evolutions} \quad \text{(IV.4)}$$

Suchperformance Hier wurden die in Tabelle IV.12 und Tabelle IV.13 gezeigten Ergebnisse gemessen.

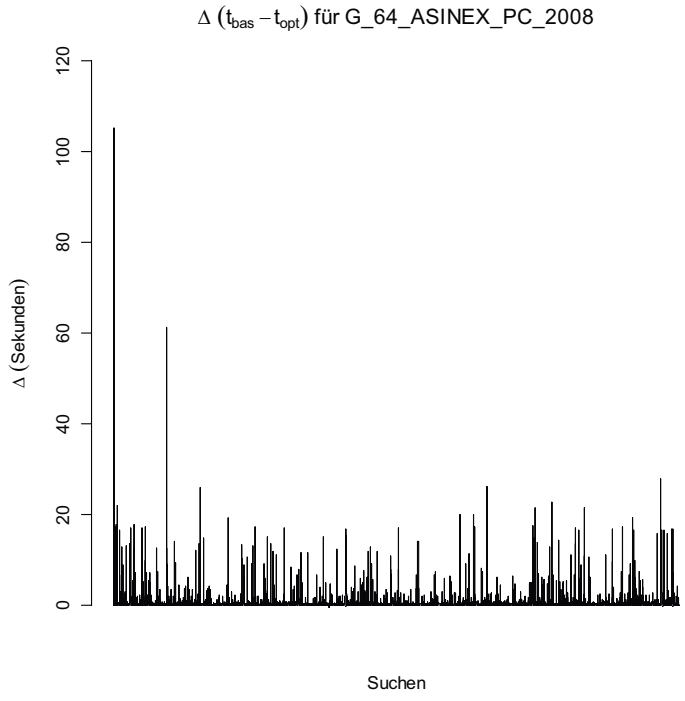
Tabelle IV.12: Ergebnisse der Optimierung für die Tabelle MAYSC mit G_64_MAYSC
Quelle: Eigene Darstellung mit R



Resultat	t_{bas} **(Sekunden)**	t_{opt} **(Sekunden)**	Δ%
$\overline{x}$	0,080	0,048	40
s	0,586	0,499	15
$\sum$	1 320,772	798,681	40
t_{INSERT}	734,073	1 086,454	-48

Resultat	n_{bas}	n_{opt}	
Verbessert			7 173
Verschlechtert			2 299
Unverändert			7 115
Δ_S			83 %
Größe der Quarantänetabelle			2 056
Dauer der Optimierung in Sek.			96 785
Suchen unterhalb der Messgrenze	6 858	7 978	

Tabelle IV.13: Ergebnisse der Optimierung für die Tabelle ASINEX_PC_2008 mit G_64_ASINEX_PC_2008
Quelle: Eigene Darstellung mit R



Resultat	t_{bas} **(Sekunden)**	t_{opt} **(Sekunden)**	Δ%
$\overline{x}$	0,394	0,219	44
s	2,793	2,003	28
$\sum$	6 536,420	3 636,211	44
t_{INSERT}	2 378,912	2 907,812	-22

Resultat	n_{bas}	n_{opt}	
Verbessert			6 124
Verschlechtert			1 905
Unverändert			8 558
Δ_S			92 %
Größe der Quarantänetabelle			17 932
Dauer der Optimierung in Sek.			1 976 400
Suchen unterhalb der Messgrenze	8 446	10 098	

IV.4.2 Lineare Optimierung

Greedy-Algorithmus

Da sich der Simplex-Algorithmus, wie in Unterabschnitt III.3.4 gezeigt, für das Problem DICTIONARY äquivalent zum Greedy-Algorithmus verhält, aber aufwändiger zu implementieren ist, wurden mit ihm keine gesonderten Versuche durchgeführt.

Selektivität Tabelle IV.14 zeigt die Ergebnisse des Versuchsplans aus Tabelle IV.3.

Für jede Behandlung wurden die Veränderung der Selektivität Δ_S und die Laufzeit t_{Run} gemessen. t_{Run} ist mit $\varnothing\ 381,307 \pm 12,951$ Sekunden im beobachteten Intervall weitgehend *konstant*. Δ_S hingegen zeigt eine Veränderung abhängig von der Anzahl der Muster, die eine Abhängigkeit vermuten lässt.

Zunächst wurden die Δ_S für 8 bis 64 Muster wie in Abbildung IV.6 graphisch dargestellt. Der Versuch, einen vermuteten Zusammenhang zwischen der Anzahl der Muster und Δ_S mittels einfacher linearer Regression mit der *Methode der kleinsten Quadrate* [RVG07, S. 154 ff.] auf ein Modell I der Regressionsanalyse [RVG07, S. 152 ff] abzubilden, führte zu dem ebenfalls in Abbildung IV.6 gezeigten Modell A. Der Determinationskoeffizient r^2 ist mit 0,931 sehr gut. Die t-Werte für β_0 und β_1 sind deutlich größer als der kritische Wert 1,943 mit $p < 0.05$ für 6 Freiheitsgrade, β_0 und β_1 haben also einen signifikanten Einfluss auf Δ_S.

Tabelle IV.14: Versuchsplan des Greedy-/LP-Algorithmus mit Beobachtungen
Quelle: Eigene Darstellung

		Factor 1	Response 1	Response 2
Std	Run	A:Wörterbuchgröße Gene	Δ_S	t_{Run} msec
1	1	8	0,578	392,482
2	2	16	0,622	372,794
3	3	24	0,644	373,169
4	4	32	0,731	389,841
5	5	40	0,737	376,731
6	6	48	0,773	367,996
7	7	56	0,790	404,997
8	8	64	0,793	372,450

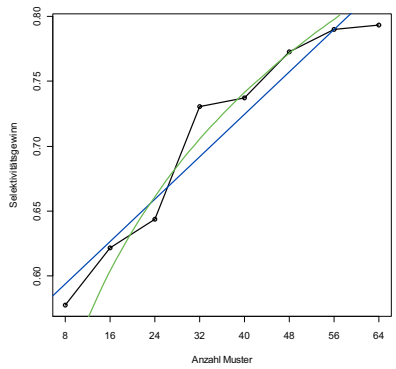
Die Modell A zugehörige Geradengleichung lautet: $\Delta_S = 0,004\,082 \times x + 0,561\,458$.

Allerdings verläuft die Kurve durch die Messpunkte augenscheinlich eher wie eine konkave Funktion mit abnehmendem Grenzertrag, ähnlich der neoklassischen Produktionsfunktion [DI94, S. 594]. Dies resultiert daraus, dass Δ_S sich definitionsgemäß nur im Intervall $[0, 1]$ bewegen kann und der Greedy-Algorithmus die Muster für das reduzierte Wörterbuch ja in absteigender Reihenfolge nach ihrem individuellen Selektivitätsertrag auswählt.

Eine solche Funktion wurde in Abbildung IV.6 beispielhaft grün eingezeichnet. Es kann daher näherungsweise von einem logarithmischen Zusammenhang zwischen der Anzahl der Muster und Δ_S ausgegangen werden.

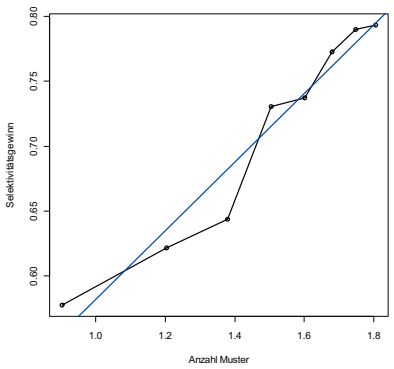
Daher wurde ein alternatives Modell B mit vorheriger $\log_{10}$ Transformation erstellt. Abbildung IV.6 zeigt, dass sich der Determinationskoeffizient r^2 mit 0,945 nochmals etwas verbessert hat. Die t-Werte für β_0 und β_1 sind deutlich grösser als der kritische Wert 1,943 (0.95, 6 DF) mit $p < 0.05$, β_0 und β_1 haben also einen signifikanten Einfluss auf Δ_S.

Modell A



$\Delta_S = \beta_0 + \beta_1 \cdot x$
$\beta_0 = 0,561\,458; t = 30,84; p = 0,000\,000\,077\,1$
$\beta_1 = 0,004\,082; t = 9,06; p = 0,000\,101$
$r^2 = 0,931\,9$
$F = 82,08$ on 1 and 6 DF, $p = 0,000\,1014$

Modell B



$\Delta_S = \beta_0 + \beta_1 \cdot \log_{10}(x)$
$\beta_0 = 0,318\,3; t = 8,187; p = 0,000\,179$
$\beta_1 = 0,263\,8; t = 10,218; p = 0,000\,051\,2$
$r^2 = 0,945\,7$
$F = 104,4$ on 1 and 6 DF, $p = 0,000\,051\,2$

Abbildung IV.6: Lineare Regression der response curve mit den zugehörigen Modellen A und B
Quelle: Eigene Darstellung

Die Modell B zugehörige transformierte Geradengleichung lautet: $\log_{10}(\Delta_S) = 0,263\,8 \times x + 0,318\,3$.

Auf Basis dieser Näherungen kann nun die theoretisch notwendige Anzahl Muster für den maximal möglichen Δ_S von 1 prognostiziert werden.

Punktprognose für Modell A:

$$\begin{aligned} 1 &= 0,004\,082 \times x + 0,561\,458 \\ x &= \frac{59\,624\,309}{554\,990} \\ x &= 107,433 \end{aligned}$$

Punktprognose für Modell B:

$$\begin{aligned} 1 &= 0,263\,8 \times \log_{10}(x) + 0,318\,3 \\ \log_{10}(x) &= \frac{6\,817}{2\,638} \\ x &= 10^{\frac{6\,817}{2\,638}} = 383,844 \end{aligned}$$

Modell A prognostiziert ein reduziertes Wörterbuch mit 107 Mustern, Modell B prognostiziert ein reduziertes Wörterbuch mit 384 Mustern, um ein maximales Δ_S zu erreichen.

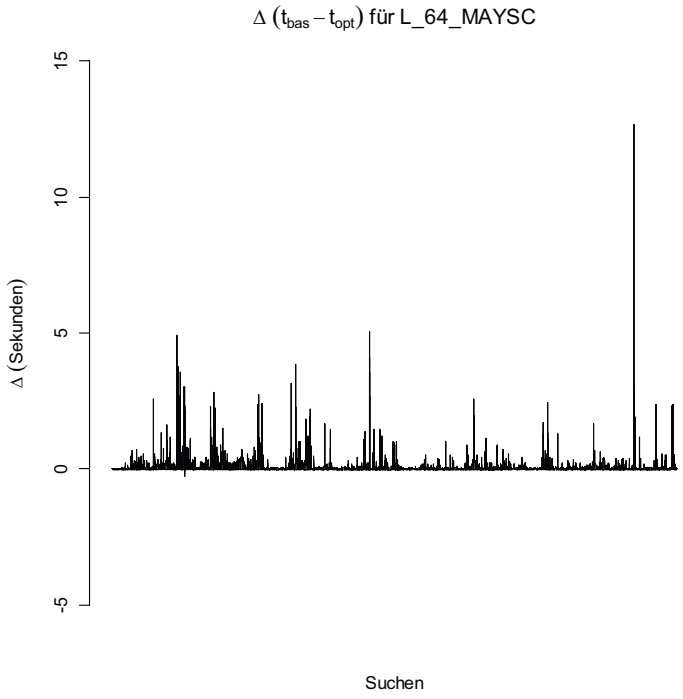
Zur Überprüfung dieser Prognosen wurde ein reduziertes Wörterbuch mit 128 Mustern erzeugt und der prognostizierte Δ_S beider Näherungen mit dem tatsächlichen Ergebnis verglichen.

Modell A prognostiziert $\Delta_S = 1$, Modell B $\Delta_S = 0,875$. Das gemessene Ergebnis ist $\Delta_S = 0,832$.

Modell B bildet den Zusammenhang zwischen der Anzahl der Muster im reduzierten Wörterbuch und Δ_S also genauer ab als Modell A.

Suchperformance Hier wurden die in Tabelle IV.15 und Tabelle IV.16 gezeigten Ergebnisse gemessen.

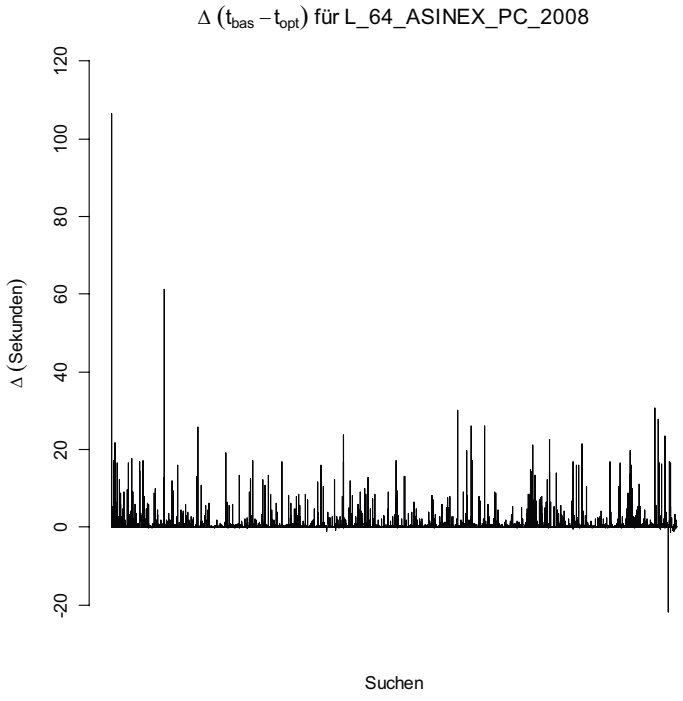
Tabelle IV.15: Ergebnisse der Optimierung für die Tabelle MAYSC mit L_64_MAYSC
Quelle: Eigene Darstellung mit R



Resultat	t_{bas} **(Sek.)**	t_{opt} **(Sek.)**	Δ%
$\overline{x}$	0,078	0,046	41
s	0,572	0,490	14
$\sum$	1 292,936	769,357	40
t_{INSERT}	734,073	944,743	-29

Resultat	n_{bas}	n_{opt}
Verbessert		6 730
Verschlechtert		2 602
Unverändert		7 255
Δ_S		79 %
Tupel in der Quarantänetabelle		2 056
Dauer der Optimierung in Sek.		372
Suchen unterhalb der Messgrenze	6 889	8 045

Tabelle IV.16: Ergebnisse der Optimierung für die Tabelle ASINEX_PC_2008 mit L_64_ASINEX_PC_2008
Quelle: Eigene Darstellung mit R



Resultat	t_{bas} **(Sekunden)**	t_{opt} **(Sekunden)**	Δ%
$\overline{x}$	0,394	0,212	46
s	2,793	1,958	30
$\sum$	6 277,620	3 520,852	44
t_{INSERT}	2 378 912	2 456 914	-3

Resultat	n_{bas}	n_{opt}	
Verbessert			5 782
Verschlechtert			2 406
Unverändert			8 399
Δ_S			93 %
Größe der Quarantänetabelle			17 932
Dauer der Optimierung in Sek.			4 158
Suchen unterhalb der Messgrenze	8 446	9 471	

IV.4.3 Optimierung auf Stichprobenbasis

Aufgrund des schlechten Laufzeitverhaltens, insbesondere der genetischen Optimierung, wurde nach einer Methode gesucht, t_{Run} zu reduzieren ohne Δ_S wesentlich zu beeinträchtigen. Dies kann grundsätzlich über zwei Methoden geschehen:

1. Änderung der freien Parameter des Algorithmus
2. Reduktion der Anzahl der Tupel in T_Q

Die zulässige Verschlechterung für Δ_S wurde auf 5 % festgelegt.

Änderung der freien Parameter des Algorithmus Mit Hilfe des Modells IV.3 und IV.4 kann eine numerische Simulation für Δ_S und t_{Run} durchgeführt werden, indem verschiedene gültige Lösungen von IV.3, welche die zulässige Verschlechterung für Δ_S nicht verletzen, in IV.4 eingesetzt werden und dann die Lösung mit der kleinsten t_{Run} ausgewählt wird.

Eine solche Simulation wurde mit DESIGN-EASE für T_B=ASINEX_PC_2008 durchgeführt, da sich hier wie in Tabelle IV.13 ersichtlich ein besonders schlechtes t_{Run} gezeigt hatte. Die beste gefundene Lösung war:

$$\Delta_S = 0,046\,773 + 0,002\,872\,84 \times 100 + 0,002\,118\,71 \times 28 + 0,006\,190\,51 \times 64 \tag{IV.5}$$

$$\begin{aligned} \Delta_S &= 0,79 \\ \log_{10}(t_{Run}) &= 4,150\,71 + 0,021\,093 \times 100 + 0,016\,054 \times 28 \\ t_{Run} &= 10^{6,709\,522\,000\,000\,001} \\ t_{Run} &= 5\,122\,972 \end{aligned} \tag{IV.6}$$

Für ein noch zulässiges Δ_S von 79 % wird eine Laufzeit von 5 123 Sekunden prognostiziert.

Reduktion der Größe der Quarantänetabelle Wie die in Unterabschnitt IV.4.1 und Unterabschnitt IV.4.2 gewonnenen Ergebnisse zeigen, hat neben den freien Parametern auch die Größe der Quarantänetabelle T_Q Einfluss auf T_{Run} (IV.7).

$$T_{Run} \in \mathcal{O}(T_Q) \tag{IV.7}$$

Die zweite Möglichkeit t_{Run} zu reduzieren besteht also darin, die Größe der Quarantänetabelle T_Q zu reduzieren. Ein dazu geeignetes Verfahren verwendet eine Zufallsstichprobe aus T_Q als neue T'_Q.

Die einfache Formel zur Berechnung der optimalen Stichprobengrösse nach Cochran [Coc72, S. 96] lautet:

$$n_0 = \frac{t^2 \cdot P \cdot Q}{d^2}, \;\; Q = 1 - P, \;\text{für}\; N \to \infty \tag{IV.8}$$

Die Stichprobengrösse wird nur von der gewünschten Genauigkeit (determiniert durch t und d) sowie dem Stichprobenanteil P bestimmt und ist von der Größe von N *unabhängig* [Coc72, S. 40].

Für ${}^{n_0}/_N \geq 5$ % von N sollte eine Endlichkeitskorrektur berücksichtigt werden [Coc72, S. 40]. Dies führt zu folgender Formel zur Berechnung der optimalen Stichprobengrösse mit Endlichkeitskorrektur nach Cochran [Coc72, S. 96]:

$$n_{korr} = \frac{\frac{t^2 \cdot P \cdot Q}{d^2}}{1 + \frac{1}{N} \cdot (\frac{t^2 \cdot P \cdot Q}{d^2} - 1)}, \;\; Q = 1 - P \tag{IV.9}$$

Die Endlichkeitskorrektur kann n_{korr} gegenüber n_0 nur verringern, nicht vergrößern, d.h. $n_{korr} \leq n_0$.

Wie Abbildung IV.7 zeigt, führt die Verwendung einer Stichprobe mit Endlichkeitskorrektur aus T_Q als neue T'_Q dazu, dass:

1. die Anzahl der Tupel in T'_Q durch eine obere Grenze limitiert wird, die kleiner ist als die Anzahl der Tupel in T_Q, N
2. die Anzahl der Tupel in T'_Q für $N < 10^5$ durch die Endlichkeitskorrektur noch weiter verringert wird

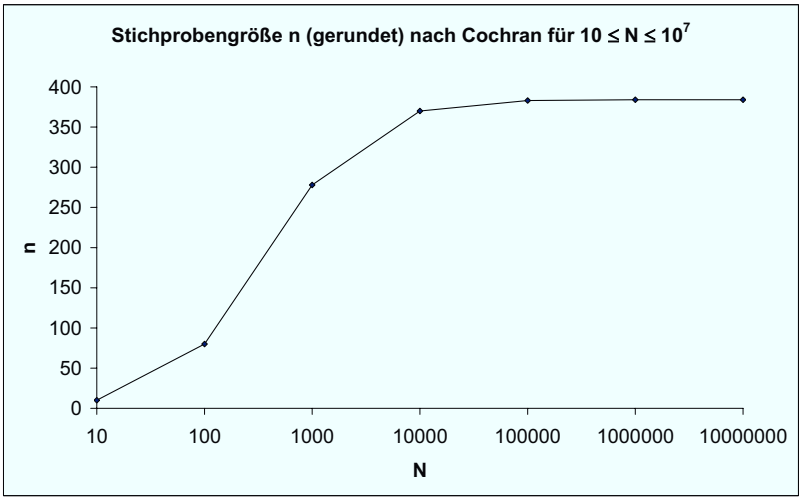


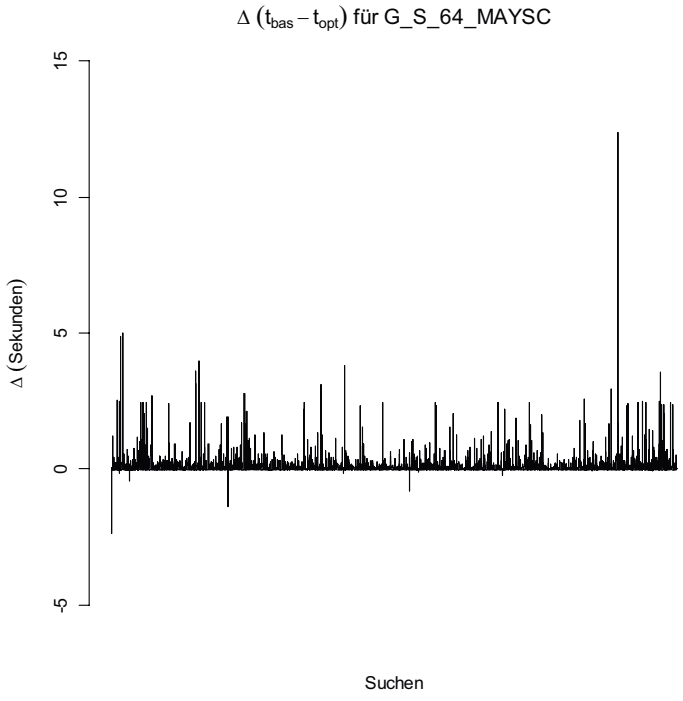
Abbildung IV.7: Optimale Stichprobengröße mit Endlichkeitskorrektur nach Cochran
Quelle: Eigene Darstellung

Da hier keine Berechnung einer Prognose des Ergebnisses möglich war, wurden die Messungen aus Unterabschnitt IV.4.1 und Unterabschnitt IV.4.2 für GA und Greedy/LP auf Basis einer Stichprobe wiederholt. Die Stichprobengrösse wurde mittels (IV.9) berechnet. Für eine Breite des Konfidenzintervalls von $\pm$ 5 % und einen unbekannten Anteil des Untersuchungsmerkmals (bei unbekanntem Anteil des Untersuchungsmerkmals wird $P = 0,5$ angenommen) in der Grundgesamtheit ergeben sich für $t = 1,96$, $P = 0,5$ und $d = 0,05$.

Für die genetische Optimierung mit T_Q auf Stichprobenbasis wurden die Ergebnisse in Tabelle IV.17 und Tabelle IV.18 gemessen.

Für die Greedy/LP Optimierung mit T_Q auf Stichprobenbasis wurden die Ergebnisse in Tabelle IV.19 und Tabelle IV.20 gemessen.

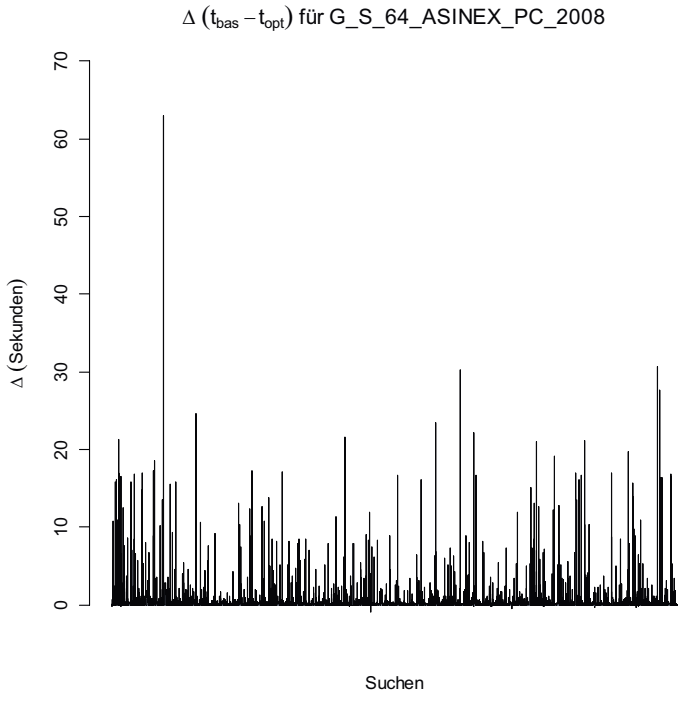
Tabelle IV.17: Ergebnisse der Optimierung für die Tabelle MAYSC mit G_S_64_MAYSC
Quelle: Eigene Darstellung mit R



Resultat	t_{bas} **(Sekunden)**	t_{opt} **(Sekunden)**	Δ%
$\overline{x}$	0,080	0,047	41
s	0,57	0,503	12
$\sum$	1 319,912	779,954	41
t_{INSERT}	734,073	1 086,454	-48

Resultat	n_{bas}	n_{opt}	
Verbessert			6 846
Verschlechtert			2 572
Unverändert			7 169
Δ_S			82 %
Größe der Quarantänetabelle			773
Dauer der Optimierung in Sek.			30 213
Suchen unterhalb der Messgrenze	6 809	7 975	

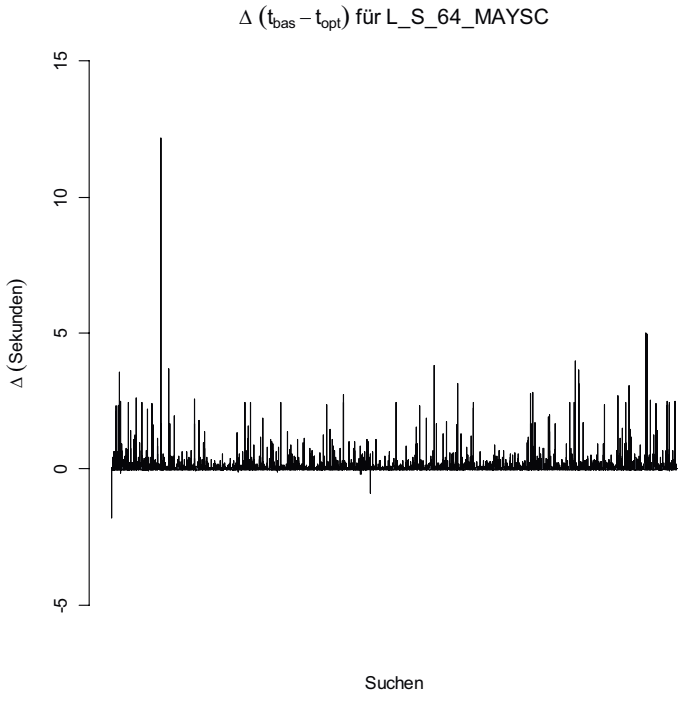
Tabelle IV.18: Ergebnisse der Optimierung für die Tabelle ASINEX_PC_2008 mit G_S_64_ASINEX_PC_2008
Quelle: Eigene Darstellung mit R



Resultat	t_{bas} **(Sekunden)**	t_{opt} **(Sekunden)**	Δ%
$\overline{x}$	0,398	0,224	44
s	2,589	1,975	24
$\sum$	6 456,523	3 711,498	43
t_{INSERT}	2 378,912	2 907,812	-22

Resultat	n_{bas}	n_{opt}	
Verbessert			5 978
Verschlechtert			2 245
Unverändert			8 364
Δ_S			90 %
Größe der Quarantänetabelle			944
Dauer der Optimierung in Sek.			43 230
Suchen unterhalb der Messgrenze	8 451	9 641	

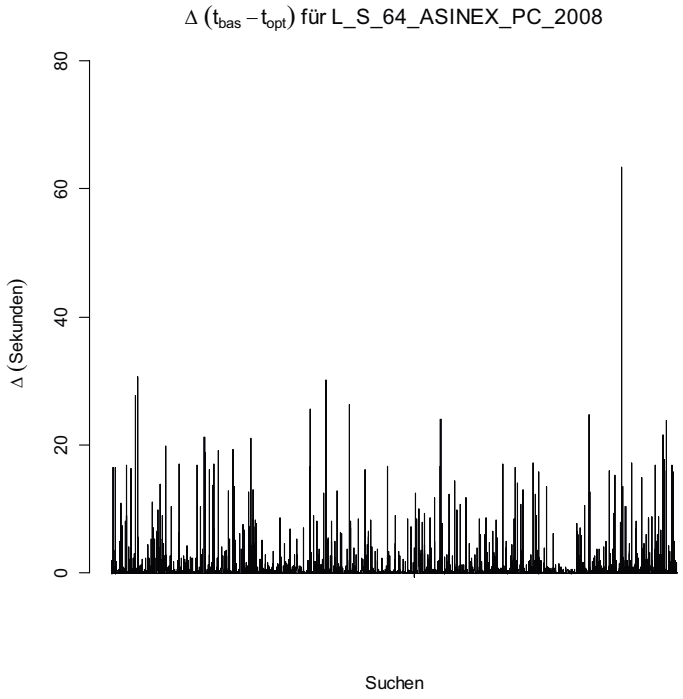
Tabelle IV.19: Ergebnisse der Optimierung für die Tabelle MAYSC mit L_S_64_MAYSC
Quelle: Eigene Darstellung mit R



Gemessen	t_{bas} **(Sek.)**	t_{opt} **(Sek.)**	Δ%
$\overline{x}$	0,080	0,046	43
s	0,570	0,498	13
$\sum$	1 319,921	768,129	42
t_{INSERT}	734,073	944,743	-29

Gemessen	**bas**	**opt**
Verbessert		6 837
Verschlechtert		2 546
Unverändert		7 204
Δ_S		78 %
Tupel in der Quarantänetabelle		796
Dauer der Optimierung in Sek.		105
Suchen unterhalb der Messgrenze	6 809	7 984

Tabelle IV.20: Ergebnisse der Optimierung für die Tabelle ASINEX_PC_2008 mit L_S_64_ASINEX_PC_2008
Quelle: Eigene Darstellung mit R



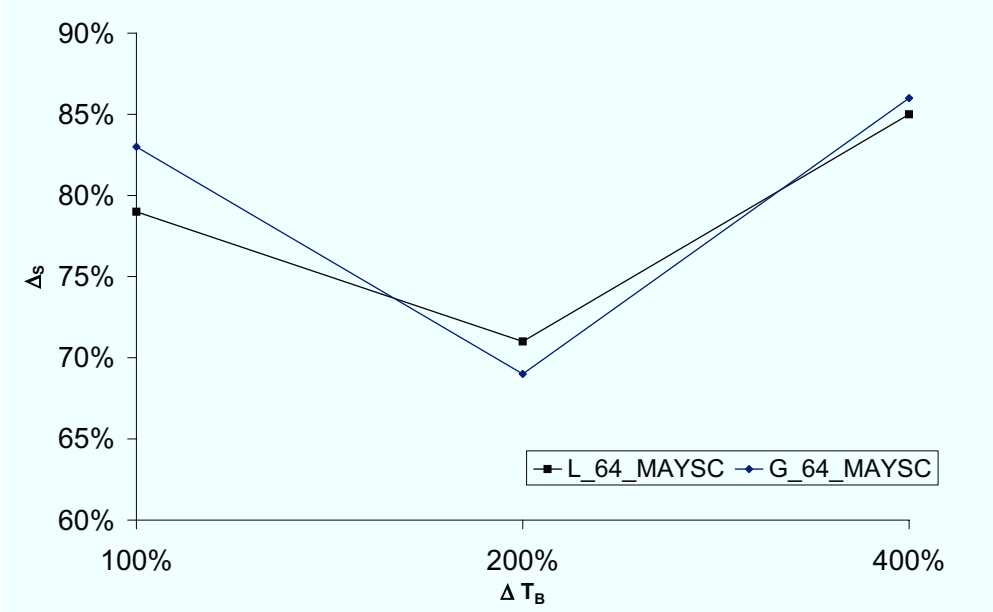
Resultat	t_{bas} **(Sekunden)**	t_{opt} **(Sekunden)**	Δ%
$\overline{x}$	0,389	0,210	46
s	2,589	1,926	26
$\sum$	6 277,620	3 520,852	44
t_{INSERT}	2 378,912	2 456 914	-3

Resultat	n_{bas}	n_{opt}	
Verbessert			5 999
Verschlechtert			2 208
Unverändert			8 380
Δ_S			92 %
Größe der Quarantänetabelle			923
Dauer der Optimierung in Sek.			186
Suchen unterhalb der Messgrenze	8 451	9 636	

IV.4.4 Änderungsstabilität des optimierten Index

Wie in Unterabschnitt IV.3.3 beschrieben, wurde zunächst MAYSC optimiert und das entsprechende reduzierte Wörterbuch erzeugt, dann relativ unspezifische Daten aus ASINEX_GC_2008 eingefügt und zum Schluss wieder ähnliche Daten aus ASINEX_PC_2008 eingefügt.

Tabelle IV.21: Änderungsstabilität bei INSERT in T_B ohne Neuoptimierung
Quelle: Eigene Darstellung



Wörterbuch	Datenquelle	T_B	Δ_S%	T_Q
L_64_MAYSC	MAYSC	58 159	79	2 056
L_64_MAYSC	ASINEX_GC_2008	116 318	71	6 950
L_64_MAYSC	ASINEX_PC_2008	232 636	85	38 625
G_64_MAYSC	MAYSC	58 159	83	2 056
G_64_MAYSC	ASINEX_GC_2008	116 318	69	7 666
G_64_MAYSC	ASINEX_PC_2008	232 636	85	43 279

Nach Abschluss jeder Einfügeoperation wurde das mit dem nicht neu optimierten reduzierten Wörterbuch erreichte Δ_S gemessen. Alle Messungen wurden sowohl für den GA, als auch für den LP Algorithmus durchgeführt.

An den in Tabelle IV.21 dargestellten Ergebnissen lassen sich, neben der schon bekannten absolut etwas besseren Optimierungsleistung des GA Algorithmus, drei interessante Beobachtungen machen:

1. Werden Daten hinzugefügt, die strukturell von den zum Zeitpunkt der Optimierung vorhandenen Daten abweichen, verschlechtert sich Δ_S
2. Werden Daten hinzugefügt, die strukturell zu den zum Zeitpunkt der Optimierung vorhandenen Daten ähnlich sind, verbessert sich Δ_S - und zwar sogar über den ursprünglich erreichten Optimierungsgrad hinaus
3. Der durch den GA Algorithmus erreichte höhere Optimierungsgrad führt dazu, dass die Verschlechterung von Δ_S durch strukturell von den zum Zeitpunkt der Optimierung vorhandenen abweichende Daten stärker ausfällt, als bei der weniger effektiven Optimierung durch den Greedy-/LP-Algorithmus.

Diskussion, Zusammenfassung und Ausblick

Every attempt to employ mathematical methods, in the study of chemical questions, must be considered profoundly irrational and contrary to the spirit of chemistry. If mathematical analysis should ever hold a prominent place in chemistry - an aberration which is happily almost impossible - it would occasion a rapid and widespread degeneration of that science. - *A. Comte*

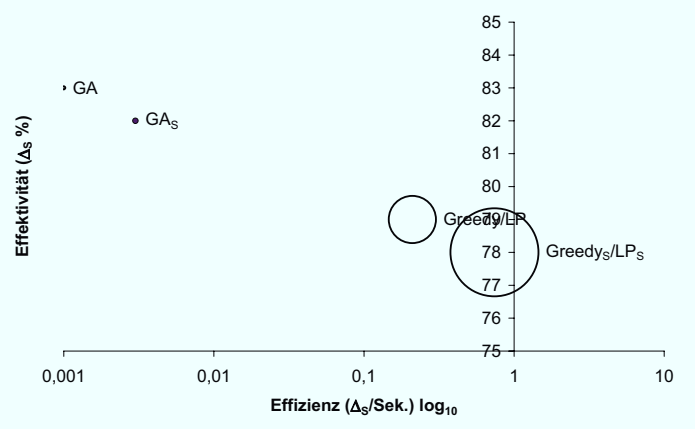
5.5 Diskussion

5.5.1 Die Güte der verwendeten Optimierungsalgorithmen

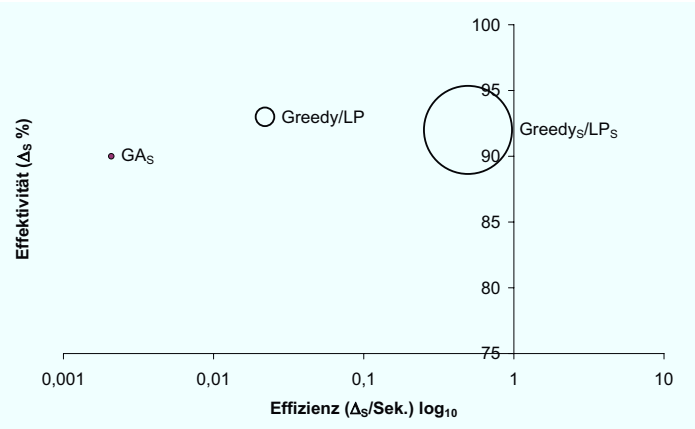
Zur einfacheren Vergleichbarkeit der experimentell gewonnenen Ergebnisse wurde eine relative Bewertung der verwendeten Optimierungsalgorithmen zueinander durchgeführt. Aus der gemessenen Effizienz, ausgedrückt als $^{\text{Selektivitätsgewinn}}/_{\text{Laufzeit}}$ und der Effektivität, ausgedrückt als maximal erreichtes Δ_S, wurde ein Gütewert als *Effizienz* $\times$ *Effektivität* ermittelt. Je höher die Güte, desto besser ist der Algorithmus zur Lösung des in Kapitel III vorgestellten Problems geeignet.

Tabelle 5.22 zeigt die Ergebnisse für die Basistabellen MAYSC und ASINEX_PC_2008, absteigend nach Güte geordnet. Die zugehörigen Blasendiagramme tragen die Effektivität auf der Ordinate, die Effizienz auf der Abszisse ab und zeigen durch die Fläche der Blase die Güte an.

Tabelle 5.22: Relative Bewertung der Optimierungsalgorithmen für MAYSC und ASINEX_PC_2008
Quelle: Eigene Darstellung



Algorithmus	Effizienz ($\%\Delta_S/t_{Run}$ Sek.)	Effektivität ($\%\Delta_S$)	Güte
$\text{Greedy}_{\text{S(tichprobe)}}/\text{LP}_{\text{S(tichprobe)}}$	0,74	78	57,720
Greedy/LP	0,210	79	16,590
$\text{GA}_{\text{S(tichprobe)}}$	0,003	82	0,246
GA	0,001	83	0,083



Algorithmus	Effizienz ($\%\Delta_S/t_{Run}$ Sek.)	Effektivität ($\%\Delta_S$)	Güte
$\text{Greedy}_{\text{S(tichprobe)}}/\text{LP}_{\text{S(tichprobe)}}$	0,495	92	45,540
Greedy/LP	0,022	93	2,046
$\text{GA}_{\text{S(tichprobe)}}$	0,002	90	0,187
GA[1]	$4{,}655 \times 10^{-5}$	92	0,004

[1] Im Diagrammmaßstab nicht mehr darstellbar

Greedy/LP führt in allen Fällen aufgrund der besseren Effizienz bei der Güte mit deutlichem Abstand vor GA, obwohl die absolut erreichte Effektivität bis zu 14 Prozentpunkte geringer ausfällt. Im Hinblick auf eine praktische Verwendung besonders interessant sind die Ergebnisse der Algorithmen, bei denen T_Q durch eine Zufallsstichprobe erzeugt wird. Bei marginaler Verschlechterung von Δ_S, verbessert sich t_{Run} von $\mathcal{O}(N)$ auf $\mathcal{O}(n);\ n < N$.

Wie in Unterabschnitt IV.4.3 bereits beschrieben wurde, nähert sich bei Berücksichtigung der Endlichkeitskorrektur die Stichprobengröße für wachsende N asymptotisch einem von N unabhängigen Maximalwert an, so dass für $N \to \infty$ sogar $\mathcal{O}(1)$ angenommen werden kann. Dies erklärt den deutlichen Effizienzvorsprung der Algorithmen auf Stichprobenbasis gegenüber den Algorithmen, die auf der Grundgesamtheit arbeiten.

Der in Unterpunkt IV.4.2 beim Greedy/LP beobachtete abnehmende Grenzertrag pro zusätzlichem Muster im reduzierten Wörterbuch folgt direkt aus der Funktionsweise des in Unterpunkt III.3.4 beschriebenen Greedy-Algorithmus, der die Muster für das reduzierte Wörterbuch ja in absteigender Reihenfolge nach ihrem individuellen Selektivitätsertrag auswählt.

5.5.2 Visualisierung des Lösungsraumes

Es kann aufgrund der Ergebnisse an dieser Stelle auch eine näherungsweise Visualisierung des Lösungsraumes des Realproblems DICTIONARY versucht werden. Um die Multidimensionalität der möglichen Lösungen auf eine geeignete zweidimensionale Darstellung abbilden zu können, wurden alle theoretisch möglichen reduzierten Wörterbücher mit n Mustern halbiert und die Hälften k_A und k_B auf der Abszisse bzw. der Ordinate abgetragen. Am Schnittpunkt befindet sich das Δ_S des Wörterbuchs $k_A \cup k_B$. Abbildung 5.8 zeigt das Ergebnis in Form einer Höhenkarte, bei der Δ_S anhand der Farbe des Schnittpunktes abgelesen werden kann.

Der vermutete Lösungsraum stellt sich als zerklüftete Topologie dar, in der eine Vielzahl verschiedener Lösungen mit ähnlichen Δ_S existieren. Ein geführter stochastischer Algorithmus, wie hier durch den GA vertreten, kommt hier erwartungsgemäß zu guten Ergebnissen, zumindest wenn man nur die Effektivität der Lösung betrachtet.

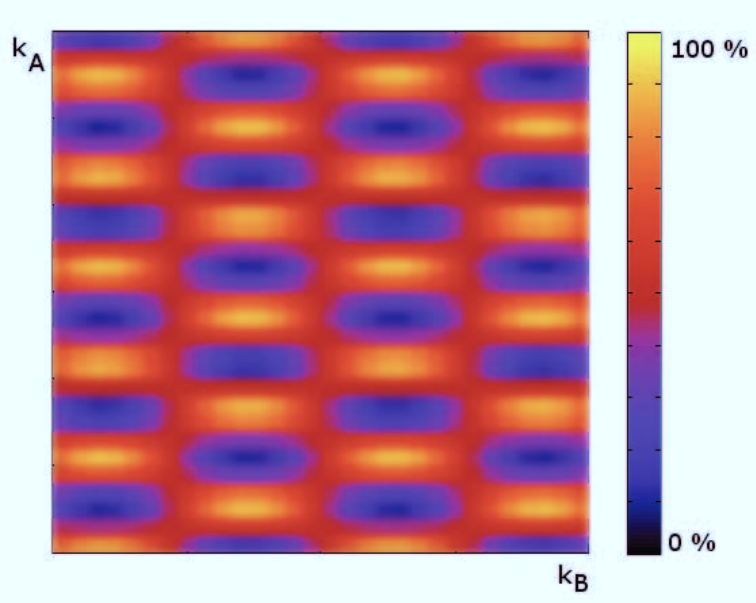


Abbildung 5.8: Vermuteter Lösungsraum für das Problem DICTIONARY
Quelle: Eigene Darstellung

Erstaunlich ist dagegen das Abschneiden des Greedy-/LP-Algorithmus, der bei der Effektivität das zweitbeste Ergebnis liefert und bei der Effizienz sogar mit deutlichem Abstand den ersten Platz belegt.

5.5.3 Die Praxistauglichkeit des entwickelten Optimierungsverfahrens

Um abschließend die Praxistauglichkeit des experimentell entwickelten Optimierungsverfahrens zu überprüfen, wurde noch ein Optimierungsversuch mit einem realen Datenbestand durchgeführt. Aus dem zum Zeitpunkt des Versuchs etwa $1,7 \times 10^6$ Strukturen umfassenden Inhalt des Laborlogistiksystems Verfügbare Chemikalien (VC) der BBS wurden per Zufallsstichprobe 10^6 verschiedene Strukturen extrahiert und in eine PGCHEM::TIGRESS Tabelle importiert. Diese Stichprobe enthält sowohl Strukturen, die aus Strukturkatalogen wie Available Chemicals Directory (ACD) [ACD09] und Available Chemicals Exchange (ChemACX) [Che09a] stammen als auch nur intern verfügbare Strukturen des Bayer Konzerns.

Da diese Daten als Betriebsgeheimnis eingestuft sind, wurde dieser Versuch intern auf einem Notebook der BBS mit Intel® Core™ Duo T2500 Mikroprozessor mit 2 GHz Taktfrequenz und 2 GB Hauptspeicher unter dem Betriebssystem Windows®

XP Professional (32 Bit, Service Pack 2) durchgeführt. Da der T2500 zum ansonsten verwendeten T2450 bis auf einen etwa zehn Prozent höheren Takt des Front Side Bus (FSB) baugleich ist (667 MHz/533 MHz), waren keine signifikanten, durch die unterschiedliche Hardware bedingten Verfälschungen der Ergebnisse zu erwarten. Für die Optimierung wurde aufgrund der Menge der Daten das in Kapitel IV effizienteste ermittelte Verfahren, die *Greedy/LP Optimierung mit Zufallsstichprobe*, gewählt.

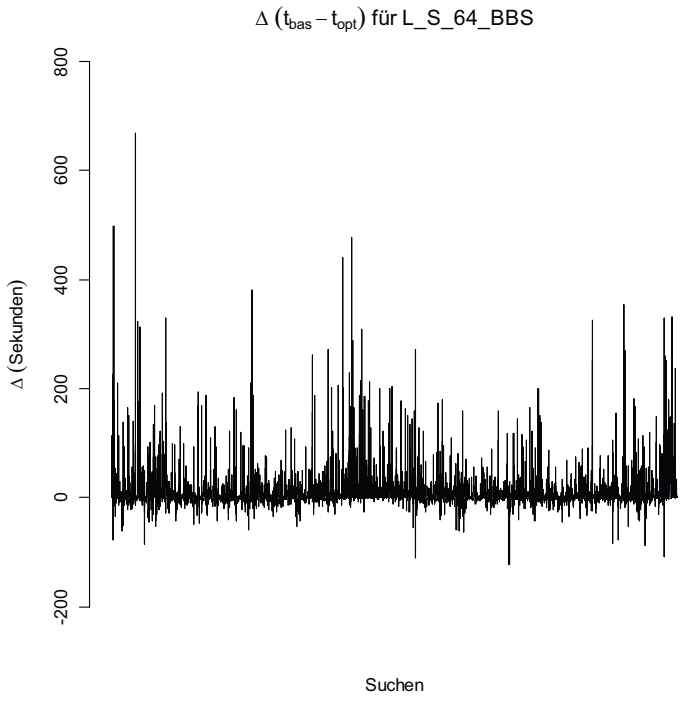
Die in Tabelle 5.23 gezeigten Ergebnisse stimmen mit einer Verbesserung der gesamten und der mittleren Suchdauer von 36 Prozent sowie einer Verbesserung der Standardabweichung der Suchdauer von 28 Prozent mit den zuvor in Kapitel IV ermittelten Resultaten überein, wobei sich zwei abweichende Beobachtungen machen lassen:

1. Die Dauer der Optimierung liegt mit etwa drei Stunden deutlich höher als in den vorausgegangenen Experimenten

2. Die absolute Anzahl von Suchen unterhalb der Messgrenze ist deutlich geringer als in den vorausgegangenen Experimenten

Diese Beobachtungen resultieren aus der in diesem abschließenden Praxisexperiment verwendeten deutlich größeren Datenmenge. Die längere Optimierungsdauer ist allerdings auch für die in Abschnitt II.2 vorgestellten Produktivsysteme nicht praxisrelevant. Insbesondere wenn man die in Unterabschnitt IV.4.4 beobachtete Änderungsstabilität des optimierten Index berücksichtigt, kann auch mit einer niedrigen Optimierungsfrequenz eine gute dynamische Optimierung gewährleistet werden.

Tabelle 5.23: Ergebnisse der Optimierung für die Tabelle BBS mit L_S_64_BBS
Quelle: Eigene Darstellung mit R



Resultat	t_{bas} **(Sekunden)**	t_{opt} **(Sekunden)**	Δ%
$\overline{x}$	9,114	5,830	36
s	35,762	25,887	28
$\sum$	151 167,900	96 709,410	36
t_{INSERT}	10 752,828	14 148,458	-24

Resultat	n_{bas}	n_{opt}	
Verbessert			8 207
Verschlechtert			7 652
Unverändert			728
Δ_S			65 %
Größe der Quarantänetabelle			1 002
Dauer der Optimierung in Sek.			11 842
Suchen unterhalb der Messgrenze	747	734	

5.6 Zusammenfassung

Das in der Einleitung definierte Ziel dieser Arbeit war, die Aufwendungen für FuE der chemischen Industrie zu reduzieren.

- direkt durch Reduktion der Kosten für den Betrieb chemischer Informationssysteme, insbesondere Katalog- und Bestellsysteme, als Teil der gesamten FuE Aufwendungen.
- indirekt durch Beschleunigung des datenbankgestützten Substanzbeschaffungsprozesses der Forschungsbereiche eines Chemieunternehmens mittels verbesserter Antwortzeiten der dort verwendeten Informationssysteme.

Durch das in dieser Arbeit entwickelte Verfahren konnte die Selektivität binärer chemischer Fingerprints um durchschnittlich 83 Prozent gesteigert sowie die mittlere Dauer einer Substruktursuche um 42 Prozent verbessert werden. Dies senkt direkt die Betriebskosten chemischer Informationssysteme durch bessere Ausnutzung vorhandener Hardware und indirekt durch die Entlastung hochqualifizierten Personals von Routinetätigkeiten wie Substanz- und Patentrecherche durch die Steigerung der Effizienz der entsprechenden unterstützenden Systeme.

Der indirekte Effekt lässt sich ohne entsprechende Feldversuche schwer quantifizieren. Für den direkten Effekt kann folgende Modellrechnung durchgeführt werden:

Tabelle 5.24: Betriebskosten für „managed Server“.
Quelle: Eigene Darstellung auf Basis der entsprechenden Angebote [1un10], [Hos10], [Het10] und [Str10]

Anbieter	**einfache Rechnerleistung** (EUR/Monat)	**doppelte Rechnerleistung** (EUR/Monat)	Δ %
1&1 Internet AG	150	300	100
Host Europe GmbH	129	249	93
Hetzner Online AG	89	149	68
Strato AG	141	201	43

In Tabelle 5.24 sind die Angebote vier etablierter Anbieter von Mietservern aufgeführt. Es zeigt sich, dass eine Verdoppelung der Rechnerleistung[1] im Mittel die Betriebskosten des Serversystems um 76 Prozent erhöht.

[1] typischerweise von Doppelkern CPU mit 4 GB Hauptspeicher auf Vierkern CPU mit 8 GB Hauptspeicher

Durch den Einsatz des in dieser Arbeit entwickelten Verfahrens kann nun, wie bereits erwähnt, die notwendige Rechenleistung zum Betrieb eines chemischen Informationssystems mit rechenintensiven Strukturen um durchschnittlich 42 Prozent reduziert werden, so dass sich durch den verbesserten Anfragedurchsatz (vgl. Definition 1 in Unterabschnitt II.2.5) der Umstieg auf die nächsthöhere verfügbare Leistungsstufe (Upgrade) eines Servers vermeiden lässt.

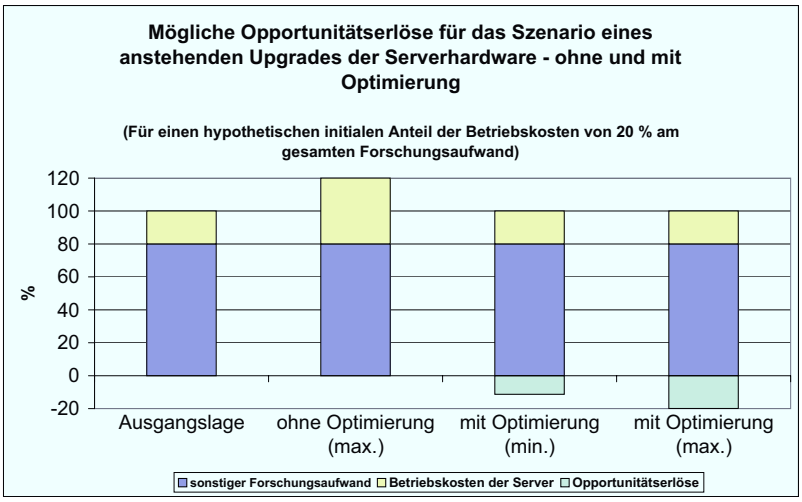


Abbildung 5.9: Mögliche Opportunitätserlöse für das Szenario eines anstehenden Upgrades der Serverhardware - ohne und mit Optimierung
Quelle: Eigene Darstellung

Zieht man Tabelle 5.24 als Bewertungsgrundlage heran, können durch den möglichen Verzicht auf ein Upgrade der Serverhardware zum Betrieb chemischer Informationssysteme Opportunitätserlöse zwischen 43 und 100 Prozent realisiert werden. Abbildung 5.9 verdeutlicht die Bandbreite möglicher Opportunitätserlöse noch einmal in graphischer Form.

Teilziele

Zur Erreichung des Ziels wurden für diese Arbeit folgende Teilziele definiert:

Beschreibung der Anwendungsgebiete und Anforderungen: Es wurden in Kapitel II die Anwendungsgebiete und Anforderungen an datenbankgestützte Informationssysteme in der forschenden chemischen Industrie vorgestellt

Beschreibung der Theorie der Verwaltung chemischer, graphischer Datentypen: Es wurde in Kapitel II und Kapitel III die notwendige Theorie für das Verständnis der Verwaltung chemischer graphischer Datentypen in RDBMS beschrieben

Beschreibung der technischen Probleme und ihrer Ursachen: Es wurden in Kapitel III die technischen Probleme solcher auf graphischen Datentypen basierenden Informationssysteme, insbesondere die suboptimale Selektivität der in der Screeningphase verwendeten Deskriptorenvektoren, mit ihren Ursachen beschrieben

Formale Beschreibung des zu lösenden Problems: Es wurde in Kapitel III eine formale Beschreibung des Problems der dynamischen kombinatorischen Optimierung von binären Deskriptorenvektoren entwickelt

Konzeption eines Verfahrens zur dynamischen kombinatorischen Optimierung: Es wurde in Kapitel III und Kapitel IV ein auf Methoden des OR basierendes Verfahren konzipiert, welches die freien Parameter der Erzeugung binärer Deskriptorenvektoren dynamisch an den konkreten Inhalt der Datenbank anpasst und so für einen gegebenen Datenbestand individuell optimiert

Nachweis der Realisierbarkeit dieses Verfahrens: Es wurde in Kapitel IV durch Implementierung einer Referenzsoftware nachgewiesen, dass das in dieser Arbeit beschriebene Verfahren realisierbar ist; seine Effektivität und Effizienz wurden durch experimentelle Messungen in Kapitel IV überprüft

Im Rahmen dieser Arbeit wurde ein Verfahren zur dynamischen Optimierung binärer chemischer Fingerprints entwickelt und beschrieben. Dieses Verfahren basiert auf Methoden der linearen und stochastischen diskreten Optimierung aus der Domäne des Operations Research (OR). Es erwies sich als geeignet, das in Kapitel III beschriebene Problem zu lösen und die in Unterabschnitt II.2.5 formulierten Anforderungen an chemische Informationssysteme zu erfüllen.

Der stetig wachsenden Verfügbarkeit chemischer Daten in Public-Domain-Datenbanken wie PubChem [Pub09b] und Chemical Entities of Biological Interest (ChEBI) [ChE09c] als auch in kommerziellen, aber öffentlich zugänglichen Katalogen wie den Maybridge Screening Compounds oder den Asinex Collections sowie kommerziellen, nicht öffentlich zugänglichen Katalogen und Datenbanken steht seit 2005 mit PGCHEM::TIGRESS ein ebenfalls freies Werkzeug gegenüber, mit dem solche Daten auch in lokalen RDBMS verwaltet werden können.

Durch die Referenzimplementierung des in dieser Arbeit entwickelten Verfahrens konnte die Suchleistung von PGCHEM::TIGRESS so gesteigert werden, dass PGCHEM::TIGRESS in allen in Abschnitt II.2 vorgestellten Systemen eingesetzt werden kann, somit eine alternative, lizenzkostenfreie Option zu kommerziellen Datenbankerweiterungen bietet und in Verbindung mit einem darauf basierenden Online-Katalogsystem (vgl. [Che09b]) einen schönen Proof-Of-Concept für dessen Praxistauglichkeit darstellt.

Natürlich kann das beschriebene Verfahren auch in jedem anderen Softwaresystem, welches mit binären Fingerprints arbeitet, Anwendung finden.

Es wurden etwa 1 000 Zeilen pl/pgSQL Code, etwa 21 000 Zeilen C und C++ Code und etwa 9 000 Zeilen Java-Code in 87 Klassen geschrieben; diese beinhalten unter anderem den GiST Index für PGCHEM::TIGRESS sowie einen neuen Typ binärer Fingerprints für OPENBABEL.

Es wurden weiterhin mindestens $5,7668487336 \times 10^{10}$ Substruktursuchen auf insgesamt 1 412 788 chemischen Strukturgraphen durchgeführt. Die für diese Arbeit entwickelte Software ist prinzipiell portierbar auf alle von POSTGRESQL und OPENBABEL unterstützten Plattformen.

5.7 Ausblick

Kommende Single Instruction Multiple Data (SIMD) Erweiterungen für Mikroprozessoren, zum Beispiel Intel® AVX mit acht 256 Bit Registern [Int09, S. 1-2] oder die Nutzung von Graphikprozessoren als generalisierte Rechenheinheiten, werden zukünftig Vergleichsoperationen auf binären Vektoren weiter beschleunigen, so dass binäre Fingerprints auf absehbare Zeit eine dominierende Technologie für das Screening chemischer graphischer Datentypen bleiben werden.

Nach bestem Wissen des Autors wurde das in dieser Arbeit entwickelte Verfahren, insbesondere unter Verwendung linearer Programme, bisher nicht in der gezeigten Weise zur Optimierung binärer chemischer Fingerprints verwendet. Nach Abschluss dieser Arbeit bleiben daher auch einige Fragen ungeklärt, die als Einstiegspunkte für weiterführende Arbeiten dienen können:

So ist das verwendete Basiswörterbuch höchstwahrscheinlich verbesserungsfähig. Optimierungen sind mindestens im Hinblick auf Menge und Art der verwendeten Substrukturmuster und die Anfälligkeit für Overtraining möglich.

Möglicherweise gibt es auch noch andere Optimierungsmethoden, die auf das Problem Dictionary erfolgreich angewendet werden können.

Des Weiteren fehlt den von Pgchem::Tigress bereitgestellten Datenbankoperatoren eine echte Selektivitätsschätzfunktion. Zurzeit werden hier nur von PostgreSQL bereitgestellte empirisch ermittelte Konstanten zurückgeliefert, um dem Optimierer überhaupt eine Schätzung zu ermöglichen.

Es bleibt außerdem zu untersuchen, ob das in dieser Arbeit vorgestellte Verfahren auch erfolgreich für das Screening anderer Graphen, zum Beispiel Verkehrszeichen oder den Globally Harmonized System of Classification and Labelling of Chemicals (GHS) Symbolen, angewendet werden können.

Literaturverzeichnis

[1un10] *Dedicated Server Übersicht*. Online. http://www.1und1.info/xml/order/ServerPremium. Version: 2010

[ACD09] *Available Chemicals Directory*. Online. http://www.symyx.com/products/databases/sourcing/acd/index.jsp. Version: 2009

[ACR09] *The CRO Market*. Online. http://www.acrohealth.org/cro-market.php. Version: 2009

[Ahr01] Ahrens, Ralph: *Neuordnung der Chemikalienpolitik in Europa. Das Weißbuch der EU-Kommission. Langfassung.* Online. http://www.vci.de/Default2~cmd~get_dwnld~docnr~88755~file~WorkshopIIIlang%2Epdf.htm. Version: 2001

[And08] Anderson, Chris: *The End of Theory: The Data Deluge Makes the Scientific Method Obsolete*. Online. http://www.wired.com/science/discoveries/magazine/16-07/pb_theory. Version: 2008

[Arb02] *Gesetz über Arbeitnehmererfindungen in der im Bundesgesetzblatt Teil III, Gliederungsnummer 422-1, veröffentlichten bereinigten Fassung, das zuletzt durch das Gesetz vom 18. Januar 2002 (BGBl. I S. 414) geändert worden ist*. Bundesministerium der Justiz, 2002

[Asi08a] *Asinex Gold Collection*. Online. http://www.asinex.com/download-zone.html. Version: 2008

[Asi08b] *Asinex Platinum Collection*. Online. http://www.asinex.com/download-zone.html. Version: 2008

[Asi08c] *Gold, Platinum & Building Blocks*. Online. http://www.asinex.com/library-gold.html. Version: 2008

[AW07] Anderson, Mark J. ; Whitcomb, Patrick J.: *DOE Simplified: Practical Tools for Effective Experimentation*. 2nd Edititon. Productivity Press New York, 2007. – ISBN 978–1–56327–344–5. – Design-Ease Version 7.2.1 P beiliegend

[Bay06] Bayer AG (Hrsg.): *Bayer-Geschäftsbericht 2006*. Bayer AG, 2006 http://www.geschaeftsbericht2006.bayer.de/de/bayer_geschaeftsbericht_2006.pdfx

[Bay07] Bayer AG (Hrsg.): *Bayer-Geschäftsbericht 2007*. Bayer AG, 2007 http://www.geschaeftsbericht2007.bayer.de/de/bayer_geschaeftsbericht_2007.pdfx

[Bay08] Bayer AG (Hrsg.): *Geschäftsbericht 2008*. Bayer AG, 2008 http://www.geschaeftsbericht2008.bayer.de/de/bayer_geschaeftsbericht_2008.pdfx

[Bäc96] Bäck, Thomas: *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, 1996. – ISBN 0–19–509971–0

[BC03] Bain & Company, Inc.: *HAS THE PHARMACEUTICAL BLOCKBUSTER MODEL GONE BUST?* Online. http://www.bain.com/bainweb/publications/printer_ready.asp?id=14243. Version: 2003

[BMG96] Bohacek, Regine S. ; McMartin, Colin ; Guida, Wayne C.: The Art and Practice of Structure-Based Drug Design: A Molecular Modeling Perspective. In: *Medicinal Research Reviews* 16 (1996), S. 43

[Bor82] Borgwardt, K.-H.: The average number of pivot steps required by the simplex method is polynomial. In: *Zeitschrift für Operations Research 26*, 1982, S. 157–177

[Bre07] Brenner, Walter: *Grundzüge des Informationsmanagements*. Springer-Verlag Berlin, 2007. – ISBN 978–3540585176

[BSAA04] Bandi, Nagender ; Sun, Chengyu ; Agrawal, Divyakant ; Abbadi, Amr E.: Hardware acceleration in commercial databases: a case study of spatial operations. In: *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, VLDB Endowment, 2004. – ISBN

0–12–088469–0, 1021–1032

[Bur09] Burjorjee, Keki M.: *Generative Fixation A Unified Explanation for the Adaptive Capacity of Simple Recombinative Genetic Algorithms*, Brandeis University, Dissertation, 2009. http://www.cs.brandeis.edu/~kekib/burjorjeeSingleSpacedDissertationAbridged.pdf. – [Online: Stand 2010-01-29T14:10:10Z]

[Che09a] *Available Chemicals Exchange*. Online. http://www.cambridgesoft.com/databases/details/?db=12. Version: 2009

[Che09b] *ChemCollect Katalog*. Online. http://www.chemcollect.de/catalog.php3. Version: 2009

[ChE09c] *Chemical Entities of Biological Interest (ChEBI)*. Online. http://www.ebi.ac.uk/chebi/. Version: 2009

[Coc72] Cochran, William G.: *Stichprobenverfahren*. Walter de Gruyter & Co., 1972. – ISBN 3110020408

[CWC71] C. West Churchman, E. Leonard A. Russel L. Ackoff A. Russel L. Ackoff: *Operations Research - Eine Einführung in die Unternehmensforschung*. 5. Auflage. R. Oldenbourg Verlag München Wien, 1971. – ISBN 978–3486434651

[Dal08] Dalke, Andrew: *Computing Tanimoto scores, quickly*. Online. http://www.dalkescientific.com/writings/diary/archive/2008/06/27/computing_tanimoto_scores.html. Version: 2008

[Dan51] Dantzig, George B.: Maximization of a linear function with subject to linear inequalities. In: *Activity Analysis of Production and Allocation*, Wiley, 1951, S. 359–373

[DE707] *Design-Ease*. Bookware: DOE Simplified: Practical Tools for Effective Experimentation, 2007. – Version 7.2.1 P

[Der09] *Derwent Innovations Index*. Online. http://thomsonreuters.com/products_services/legal/legal_products/intellectual_property/Derwent_Innovations_Index. Version: 2009

[DI94] Dichtl, Erwin (Hrsg.) ; Issing, Otmar (Hrsg.): *Vahlens Großes Wirtschaftslexikon in vier Bänden*. Bd. 1. 2. Auflage. Deutscher Taschenbuchverlag, München, 1994. – ISBN 3–423–59006–8

[DK92] Dürr, Walter ; Kleibohm, Karl: *Operations Research: Lineare Modelle und ihre Anwendungen*. 3. Auflage. Carl Hanser Verlag München Wien, 1992. – ISBN 3–446–17335–8

[DLHN02] Durant, Joseph L. ; Leland, Burton A. ; Henry, Douglas R. ; Nourse, James G.: Reoptimization of MDL Keys for Use in Drug Discovery. In: *Journal of Chemical Information and Computer Sciences* 42 (2002), S. 1273–1280

[Dor01] Dornbusch, Sascha: *Strategische Optionen in der Chemieindustrie im Kontext des E-Business*, Gerhard-Mercator-Universität Duisburg, Dissertation, 2001. `http://purl.oclc.org/NET/duett-01082002-193646`. – [Online: Stand 2009-11-10T09:45:12Z]

[EBR+95] Eriksson, Tommy ; Bjorkman, Sven ; Roth, Bodil ; Fyge, Åsa ; Höglund, Peter: Stereospecific Determination, Chiral Inversion In Vitro and Pharmacokinetics in Humans of the Enantiomers of Thalidomide. In: *Chirality* 7 (1995), S. 44–52

[Ecl09] Eclipse Foundation: *Eclipse*. Online. `http://www.eclipse.org/`. Version: 2009

[EIN02] *European INventory of Existing Commercial Chemical Substances*. Online. `http://ecb.jrc.ec.europa.eu/esis-pgm/einecs_IS_pgm.php?LANG=de&PGM=ein`. Version: 1990, 2002

[ELI09] *European LIst of Notifieded Chemical Substances*. Online. `http://ecb.jrc.ec.europa.eu/DOCUMENTS/New-Chemicals/ELINCS_PUBLICATION/ELINCS_2009.xls`. Version: 2009

[Evd08] Evdokimov, Sergei: *Secure Outsourcing of IT Services in a Non-Trusted Environment*, Humboldt-Universität zu Berlin, Dissertation, 2008. `http://deposit.d-nb.de/cgi-bin/dokserv?idn=991171357&dok_var=d1&dok_ext=pdf&filename=991171357.pdf`. – [Online: Stand 2009-08-03T13:15:30Z]

[FuE07] Wissenschaftsstatistik gGmbH (Hrsg.): *FuE Datenblatt Chemie*. Online. http://www.stifterverband.info/statistik_und_analysen/publikationen/fue_facts/fue_facts_chemie_04_2007.pdf. Version: 2007

[GE03] Gasteiger, J. (Hrsg.) ; Engel, T. (Hrsg.): *Chemoinformatics: a Textbook*. New York, NY, USA : Wiley, 2003. – ISBN 3–527–30681–1

[GJ79] Garey, Michael R. ; Johnson, David S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman and Company, 1979. – ISBN 978–0716710455

[Gol89] Goldberg, David E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman, 1989. – ISBN 978–0201157673

[Gre09] Grela, Karol: *LabJ - Electronic Laboratory Notebook, ELN*. Online. http://karolgrela.eu/labj/. Version: 2009

[Guh05] Guha, Rajarshi: *METHODS TO IMPROVE THE RELIABILITY, VALIDITY AND INTERPRETABILITY OF QSAR MODELS*, The Pennsylvania State University, Dissertation, 2005. http://edoc.hu-berlin.de/docviews/abstract.php?id=26966. – [Online: Stand 2008-04-24T08:10:20Z]

[Gut84] Guttman, Antonin: R-Trees: A Dynamic Index Structure for Spatial Searching. In: Yormark, Beatrice (Hrsg.): *SIGMOD'84, Proceedings of Annual Meeting, Boston, Massachusetts, June 18-21, 1984*, ACM Press, 1984, S. 47–57

[Hai09a] Haider, Norbert: *checkmol/matchmol*. Online. http://merian.pch.univie.ac.at/~nhaider/cheminf/cmmm.html. Version: 2009

[Hai09b] Haider, Norbert: *Creating a Web-based, Searchable Molecular Structure Database Using Free Software*. Online. http://merian.pch.univie.ac.at/~nhaider/cheminf/moldb.html. Version: 2009

[Het10] *Managed Server Produktmatrix*. Online. http://www.hetzner.de/de/hosting/produktmatrix/managed-server-produktmatrix/Produktmatrix. Version: 2010

[HNP95] Hellerstein, Joseph M. ; Naughton, Jeffrey F. ; Pfeffer, Avi: Generalized Search Trees for Database Systems. In: Dayal, Umeshwar (Hrsg.) ; Gray, Peter M. D. (Hrsg.) ; Nishio, Shojiro (Hrsg.): *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland*, Morgan Kaufmann, 1995. – ISBN 1–55860–379–4, 562-573

[Hol92] Holland, John H.: *Adaptation in natural and artificial systems*. Cambridge, MA, USA : MIT Press, 1992. – ISBN 0262581116

[Hos10] *Dedicated Server - Dedicated Server Managed*. Online. `http://www.hosteurope.de/produkte/Dedicated-Server-Managed`. Version: 2010

[HP00] Hertzberg, R. P. ; Pope, A. J.: High-throughput screening: new technology for the 21st century. In: *Curr Opin Chem Biol* 4 (2000), August, Nr. 4, 445–451. `http://view.ncbi.nlm.nih.gov/pubmed/10959774`. – ISSN 1367–5931

[Inf09] Infochem GmbH: *ICEdit*. Online. `http://www.infochem.de/en/products/software/icedit.shtml`. Version: 2009

[Int00] International Organization for Standardization: *General requirements for the competence of calibration laboratories*. Bd. ISO/IEC 17025. International Organization for Standardization, 2000

[Int07] International Union of Pure and Applied Chemistry: *The IUPAC International Chemical Identifier (InChI)*. Online. `http://www.iupac.org/inchi/release102.html`. Version: 2007

[Int09] Intel Corporation: Intel Advanced Vector Extensions Programming Reference / Intel Corporation. Version: 2009. `http://software.intel.com/file/10069`. 2009 (319433-005). – Forschungsbericht

[JCh09] *JChemPaint*. Online. `http://sourceforge.net/projects/jchempaint/`. Version: 2009

[Jim04] Jim Shore: Fail Fast. (2004). `http://www.martinfowler.com/ieeeSoftware/failFast.pdf`

[Kar72] Karp, Richard M.: Reducibility Among Combinatorial Problems. In: *Complexity of Computer Computations, Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y.. New York: Plenum*, 1972, S. 85–103

[KM72] Klee, V. ; Minty, G. J.: How good is the simplex algorithm? In: *Inequalities III*, Academic Press New York, 1972, S. 159–172

[Knu98] Knuth, Donald E.: *The Art of Computer Programming*. Bd. 3 Sorting and Searching. Second Edition. Addison-Wesley, 1998. – ISBN 0–201–89685–0

[KR01] Kanth, Kothuri Venkata R. ; Ravada, Siva: Efficient Processing of Large Spatial Queries Using Interior Approximations. In: *SSTD '01: Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases*. London, UK : Springer-Verlag, 2001. – ISBN 3–540–42301–X, S. 404–424

[Kra08] Kraut, Hans: *RE: Unterstützung für Forschungsarbeit?* 2008. – [E-Mail; Stand 17. Juni 2008]

[Kuh09] Kuhn, Thomas: *Open Source Workflow Engine for Cheminformatics: From Data Curation to Data Analysis*, Universität zu Köln, Dissertation, 2009. `http://kups.ub.uni-koeln.de/volltexte/2009/2660`. – [Online: Stand 2009-07-31T06:05:20Z]

[KV05] Korte, Bernhard ; Vygen, Jens: *Combinatorial Optimization: Theory and Algorithms*. 3rd Edition. Springer-Verlag Berlin Heidelberg New York, 2005. – ISBN 978–3–540–25684–7

[LG07] Leach, Andrew R. ; Gillet, Valerie J.: *An Introduction to Chemoinformatics*. Springer Publishing Company, Incorporated, 2007. – ISBN 1402062907, 9781402062902

[LJH05] Lutz J. Heinrich, Franz L.: *Informationsmanagement*. 8. Auflage. Oldenbourg, 2005. – ISBN 978–3486577723

[Mal97] Malcolm J. McGregor and Peter V. Pallai: Clustering of Large Databases of Compounds: Using the MDL "Keysäs Structural Descriptors. In: *Journal of Chemical Information and Computer Sciences* 37 (1997), Nr. 3, S. 443–448

[Max09] *Maxima*. Online. http://maxima.sourceforge.net/. Version: 2009. – Version 5.18.0

[May08a] *MAYBRIDGE REFERENCE HANDBOOK 45TH ANNIVERSARY EDITION*. Thermo Fisher Scientific Inc., 2008

[May08b] *Maybridge Screening Collection*. Online. http://www.maybridge.com/portal/alias__Rainbow/lang__en/tabID__146/DesktopDefault.aspx. Version: 2008

[Mic] MICROSYSTEMS, Sun: *Java SE*. Online. http://java.sun.com/javase/

[Mil68] MILLER, Robert B.: Response time in man-computer conversational transactions. In: *AFIPS '68 (Fall, part I): Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. New York, NY, USA : ACM, 1968, S. 267–277

[MM92] MÜLLER-MERBACH, Dr. H.: *Operations Research*. 3. Auflage. Verlag Franz Vahlen München, 1992. – ISBN 3–8006–0388–8

[MMV+09] MEFFERT, Klaus ; MESKAUSKAS, Audrius ; VOS, Jerry ; ROTSTANA, Neil ; MESEGUER, Javier ; MARTÍ, Enrique D.: *JGAP - Java Genetic Algorithms Package*. Online. http://jgap.sourceforge.net/. Version: 2009

[MN98] MATSUMOTO, Makoto ; NISHIMURA, Takuji: Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. In: *ACM Trans. Model. Comput. Simul.* 8 (1998), Nr. 1, S. 3–30. http://dx.doi.org/http://doi.acm.org/10.1145/272991.272995. – DOI http://doi.acm.org/10.1145/272991.272995. – ISSN 1049–3301

[Mor08] MORLEY, Chris: *Re: [Open Babel] Pure curiosity: Where do the zeroes in FP2 fragments come from?* 2008. – [E-Mail; Stand 23. Juni 2008]

[MSN06] MONDAL, Rajib ; SHAH, Bipin K. ; NECKERS, Douglas C.: Photogeneration of Heptacene in a Polymer Matrix. In: *Journal of the American Chemical Society* 128 (2006), Nr. 30, 9612-9613. http://dx.doi.org/10.1021/ja063823i. – DOI 10.1021/ja063823i

[MT90] MARTELLO, Silvano ; TOTH, Paolo: *Knapsack Problems: Algorithms and Computer Implementations*. Revised (November 1990) Edition. John Wiley

and Sons Inc., 1990. – ISBN 978–0471924203

[Mun09] Munos, Bernard: Lessons from 60 years of pharmaceutical innovation. In: *Nature Reviews Drug Discovery* 8 (2009), S. 959–968. http://dx.doi.org/http://dx.doi.org/10.1038/nrd2961. – DOI http://dx.doi.org/10.1038/nrd2961

[Nov08] Novartis AG (Hrsg.): *Geschäftsbericht 2008 der Novartis Gruppe*. Novartis AG, 2008 http://www.novartis.de/presse_bereich/downloads_broschueren/unternehmensbroschueren/NovAR08-web-D.pdf

[Ope09] *OpenBabel*. Online. http://www.openbabel.org. Version: 2009. – Version 2.2.x

[Pat08] *Patentgesetz in der Fassung der Bekanntmachung vom 16. Dezember 1980 (BGBl. 1981 I S. 1), das zuletzt durch Artikel 83a des Gesetzes vom 17. Dezember 2008 (BGBl. I S. 2586) geändert worden ist*. Bundesministerium der Justiz, 2008

[PPA05] Payne, Marcia M. ; Parkin, Sean R. ; Anthony, John E.: Functionalized Higher Acenes: Hexacene and Heptacene. In: *Journal of the American Chemical Society* 127 (2005), Nr. 22, 8028-8029. http://dx.doi.org/10.1021/ja051798v. – DOI 10.1021/ja051798v

[PPR93] Panico, R. (Hrsg.) ; Powell, W. H. (Hrsg.) ; Richer, J. C. (Hrsg.): *A Guide to IUPAC Nomenclature of Organic Compounds, Recommendations 1993*. Blackwell Scientific Publications, 1993. – ISBN 0–632–03488–2

[Pub09a] National Center for Biotechnology Information (Hrsg.): *PubChem Substructure Fingerprint V1.3*. Online. ftp://ftp.ncbi.nlm.nih.gov/pubchem/specifications/pubchem_fingerprints.pdf. Version: 2009

[Pub09b] *The PubChem Project*. Online. http://pubchem.ncbi.nlm.nih.gov/. Version: 2009

[R D09] R Development Core Team: *R: A Language and Environment for Statistical Computing*. http://www.R-project.org. Version: 2009

[RVG07] Rasch, Dieter ; Verdooren, Leon R. ; Gowers, Jim I.: *Grundlagen der Planung und Auswertung von Versuchen und Erhebungen*. 2. Auflage. R. Oldenbourg Verlag München Wien, 2007. – ISBN 978–3–486–58300–7

[SB07] Swamidass, S J. ; Baldi, Pierre: Mathematical Correction for Fingerprint Similarity Measures to Improve Chemical Retrieval. In: *Journal of Chemical Information and Modeling* (2007), April. http://dx.doi.org/10.1021/ci600526a. – DOI 10.1021/ci600526a. – ISSN 1549–9596

[Sch04] Schwister K. et. al. ; Schwister, Karl (Hrsg.): *Taschenbuch der Chemie*. 3. Auflage. Hanser Fachbuchverlag, 2004. – ISBN 978–3446228412

[SH05] Stahlknecht, Peter ; Hasenkamp, Ulrich: *Einführung in die Wirtschaftsinformatik*. 11. Auflage. Springer-Verlag Berlin Heidelberg New York, 2005. – ISBN 3–540–01183–8

[SKW00] Steinbeck, C. ; Krause, S. ; Willighagen, E.: JChemPaint - Using the Collaborative Forces of the Internet to Develop a Free Editor for 2D Chemical Structures. In: *Molecules* 5 (2000), S. 93–98

[SS09] Symyx Solustions, Inc.: *Symyx Notebbok for Synthesis*. Online. http://www.symyx.com/products/software/lab-notebooks/notebook-synthesis/index.jsp. Version: 2009

[Sti08] Stifterverband für die Deutsche Wissenschaft (Hrsg.): *FuE-Datenreport 2008 Analysen und Vergleiche*. 2008 http://www.stifterverband.info/statistik_und_analysen/publikationen/fue_datenreport/fue_datenreport_2008.pdf

[Str10] *STRATO Pro - Lösungen für Geschäftskunden - Managed Server*. Online. http://www.strato-pro.de/SITEFORUM?t=/contentManager/selectCatalog&e=UTF-8&i=1207058582423&l=1&ParentID=1235565889899&startRow=0&active=no. Version: 2010

[Sym07] Symyx Technologies, Inc.: *CTFile Formats*. Online. http://www.symyx.com/downloads/public/ctfile/ctfile.pdf. Version: 2007

[TCG93] Testa, Bernard ; Carrupt, Pierre-Alain ; Gal, Joseph: The So-called "Interconversion" of Stereoisomeric Drugs: An Attempt at Clarification. In: *Chirality* 5 (1993), S. 105–111

[tea09] TEAM, MinGW: *MinGW | Minimalist GNU for Windows*. Online. `http://www.mingw.org/`. Version: 2009

[Wei88] WEININGER, David: SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. In: *Journal of Chemical Information and Computer Sciences* 28 (1988), Nr. 1, S. 31–36

[Wei09] WEISSTEIN, Eric W.: *Dirichlet's Box Principle. From MathWorld–A Wolfram Web Resource.* Online. `http://mathworld.wolfram.com/DirichletsBoxPrinciple.html`. Version: 2009

[Wik08] WIKIPEDIA: *Nomenklatur (Chemie) — Wikipedia, Die freie Enzyklopädie*. `http://de.wikipedia.org/w/index.php?title=Nomenklatur_%28Chemie%29&oldid=48183121`. Version: 2008. – [Online; Stand 29. Juli 2008]

[Wil99] WILHELM F. MAIER: Kombinatorische Chemie – Herausforderung und Chance für die Entwicklung neuer Kataylsatoren und Materialien. In: *Angewandte Chemie* 111 (1999), S. 1294

[WWW89] WEININGER, David ; WEININGER, Arthur ; WEININGER, Joseph L.: SMILES. 2. Algorithm for generation of unique SMILES notation. In: *Journal of Chemical Information and Computer Sciences* 29 (1989), Nr. 2, S. 97–101

A Tanimoto Koeffizient und Soergel Distanz

Der Tanimoto Koeffizient C_T (A.1) ist ein Maß für die Ähnlichkeit zweier Mengen binärer Attribute $\vec{A}$ und $\vec{B}$, repräsentiert durch eine reelle Zahl im Intervall $[0, 1]$. Die Soergel Distanz D_S ist analog dazu ein Maß für den Ähnlichkeitsabstand zweier Mengen binärer Attribute.

$$C_T(\vec{A}, \vec{B}) = \frac{\vec{A} \cap \vec{B}}{\vec{A} \cup \vec{B}} \tag{A.1}$$

$$D_S(\vec{A}, \vec{B}) = 1 - C_T(\vec{A}, \vec{B}) \tag{A.2}$$

Für Mengen binärer Werte sind Tanimoto Koeffizient und Soergel Distanz komplementär (A.2).

Die in PGCHEM::TIGRESS verwendete Funktion zur Ermittlung von C_T benutzt eine Look-Up-Table in Form eines assoziativen Arrays und entspricht dem von Andrew Dalke in seinem Blog [Dal08] veröffentlichten Code *8-bit LUT v2*.

B Fragmenterzeugung des OpenBabel FP2 Algorithmus

„

```
> Hi all,
>
> out of curiosity I took a look at the fragments generated by FP2 for
> Benzene:
>
> 0|6|5|6
> 0|6|5|6|5|6
> 0|6|5|6|5|6|5|6
> 0|6|5|6|5|6|5|6|5|6
> 0|6|5|6|5|6|5|6|5|6|5|6
> 5|6|5|6|5|6|5|6|5|6|5|6
>
> If I understand the format correctly, it is an alternating sequence of
> atom and bond codes. Where do the leading zeroes come from?

For linear fragments the number of bonds is one less than the number of
atoms. So the order of this non-existent bond is set to 0. For rings the
number of bonds is equal to the number of atoms. The last two in the
list are the linear and ring forms of a fragment with six aromatic atoms.

This fingerprint code is getting a lot of detailed attention!

Chris
```

„ [Mor08]

C Spezielle Anforderungen an die Erkennung chemischer Graphen

Neben der graphentheoretischen Erkennung von Strukturgraphen, die in Unterabschnitt III.1.2 beschrieben ist, existieren weitere spezielle Anforderungen an die Erkennung von Strukturgraphen.

Sie resultieren hauptsächlich aus Synonymie in der graphischen Notation, aus der Schwierigkeit der algorithmischen Erkennung gewisser chemischer Eigenschaften sowie der Tatsache, dass chemische Verbindungen mit verschiedenen Graphen im Verständnis des Chemikers trotzdem „gleich“ sein können.

Dies liegt in der quantenchemischen Natur von Molekülen begründet, die keine Notation vollständig abbilden kann.

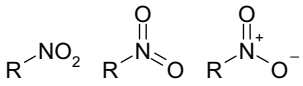
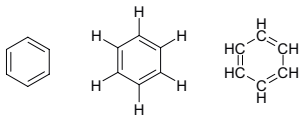
Einige folgende Beispiele mögen dies verdeutlichen. Einen umfassenderen Einstieg in die Thematik bieten die Bücher von Gasteiger und Engel [GE03] oder Leach und Gillet [LG07].

C.1 Synonymie in der graphischen Notation

Die graphische Notation chemischer Strukturen erlaubt die Verwendung unterschiedlicher Schreibweisen für dieselben Merkmale.

Für die automatisierte Erkennung werden synonyme Darstellungen, wie sie in Tabelle C.1 gezeigt werden, typischerweise in eine interne kanonische Form umgewandelt, bevor der Strukturgraph dem eigentlichen Erkennungsalgorithmus übergeben wird.

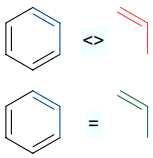
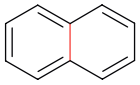
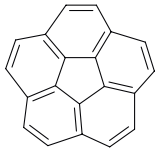
Tabelle C.1: Spezielle Anforderungen an die Erkennung von Strukturgraphen - Einfache Variationen
Quelle: Eigene Darstellung

Typ	**Beispiel**	**Problem**
Nitrogruppen		Nitrogruppen können in drei Varianten gezeichnet werden. Die mittlere Variante ist dabei tatsächlich falsch da kein Stickstoff mit fünf Bindungen existiert. Trotzdem wird sie häufig, möglicherweise wegen des geringeren Aufwands, so gezeichnet.
Implizite und explizite Wasserstoffatome		Strukturformeln werden der Übersichtlichkeit halber oft ohne Wasserstoffatome als so genannte Skelettformel gezeichnet. Werden die Wasserstoffatome explizit gezeichnet, sind zwei Varianten möglich.

C.2 Die algorithmische Erkennung chemischer Eigenschaften

Die exakte algorithmische Erkennung mancher chemischer Eigenschaften der Verbindung aus ihrem Strukturgraphen zeigt oft ein so schlechtes Laufzeitverhalten, dass auf weniger genaue Heuristiken zurückgegriffen wird. Solche Heuristiken wiederum erfassen nicht alle möglichen Fälle, so dass sie durch Listen bekannter Ausnahmen ergänzt werden müssen.

Tabelle C.2: Spezielle Anforderungen an die Erkennung von Strukturgraphen - Aromatizität
Quelle: Eigene Darstellung

Typ	Beispiel	Problem
Suche aromatischer Bindungen		Soll Aromatizität bei einer Suche berücksichtigt werden, dürfen Einfach- und Doppelbindungen nicht als äquivalent zu aromatischen Bindungen erkannt werden.
Polyzyklische aromatische Kohlenwasserstoffe (PAK)		Polyzyklische aromatische Kohlenwasserstoffe (PAK) wie Naphthalin werden von der Hückel-Regel nicht erfasst.
Ausnahmefälle		Exoten wie Corannulen, welches mit 20 π-Elektronen, fehlender Planarität und als polyzyklische Verbindung die Hückel-Regel in allen drei Punkten nicht erfüllt, de facto aber ein Aromat ist, müssen als Ausnahme erkannt werden.

Die *Aromatizität* (Definition 9) organischer Verbindungen ist beispielsweise so eine schwierig zu erkennende chemische Eigenschaft, die sich exakt nur durch quantenmechanische Verfahren ermitteln lässt. Tabelle C.2 zeigt einige der Probleme, die die algorithmische Erkennung von Aromatizität bereitet.

Definition 9. *„Aromatische Verbindungen sind cyclische, voll konjugierte Polyene, die ein mesomeriestabilisiertes π-Elektronensystem enthalten. Liegt ein solches System in einer organischen Verbindung (zum Beispiel Benzen) vor, so sind damit bestimmte charakteristische Eigenschaften verbunden.“[Sch04, S. 471]*

Daher wird als Heuristik die Rückel-Regel (Regel 1) verwendet, die allerdings nur für Monozyklen, also Verbindungen mit genau einem Ring, gilt. Um auch Polyzyklische aromatische Kohlenwasserstoffe (PAK) korrekt zu erkennen, finden zusätzliche Metaheuristiken Anwendung.

Entweder man zerlegt den Strukturgraphen in sein Smallest Set of Smallest Rings (SSSR) und bewertet jeden Ring einzeln - sind alle Ringe aromatisch, ist auch der PAK aromatisch - oder man findet den umschließenden Ring im Set of All Rings (SAR), und wenn dieser aromatisch ist, ist, unter der Annahme dass die Zentralbindungen nicht am System teilnehmen, auch der PAK aromatisch.

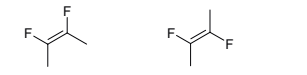
Regel 1. *Ein* monozyklisches, planares *und vollständig über den Ring* konjugiertes *(alle Ringatome sind sp^2-hybridisiert) Molekül mit* $4 \cdot n + 2, n \in \mathbb{N}_0$ *π-Elektronen ist eine aromatische Verbindung.*

C.3 Die Unschärfe des Begriffs der chemischen Gleichheit

Haben chemische Verbindungen gleiche Summenformeln, aber unterschiedliche Konformationsformeln, so spricht man von *Isomerie*, die Verbindungen sind dann *Isomere*.

Isomere zeigen unterschiedliche chemische, physikalische und physiologische Eigenschaften, so dass sie normalerweise als verschiedene Strukturen behandelt und voneinander unterschieden werden.

Tabelle C.3: Spezielle Anforderungen an die Erkennung von Strukturgraphen - Stereoisomerie
Quelle: Eigene Darstellung

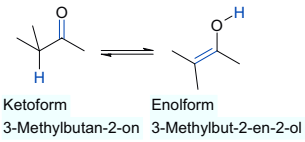
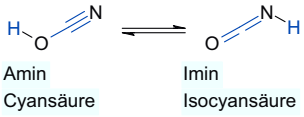
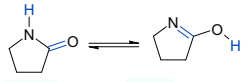
Typ	**Beispiel**	**Problem**
Z-E-Isomerie	 (2Z)-2,3-Difluorbut-2-en (2E)-2,3-Difluorbut-2-en	Soll Z-E-Isomerie bei einer Suche *nicht* berücksichtigt werden, muss die zentrale Doppelbindung um 180 ° rotiert werden, obwohl Mehrfachbindungen nicht rotierbar sind.
Cis-Trans-Isomerie	 cis-4a,8a-Dimethyldecahydronaphthalen  trans-4a,8a-Dimethyldecahydronaphthalen	Soll Cis-Trans-Isomerie bei einer Suche *nicht* berücksichtigt werden, müssen beide Varianten als äquivalent erkannt werden, obwohl die räumliche Anordnung der randständigen Bindungen relativ zur Hauptebene des Moleküls unterschiedlich ist.
R-S-Isomere (Enantiomere)	 (S)-Brom(chlor)fluormethan  (R)-Brom(chlor)fluormethan	Soll R-S-Isomerie bei einer Suche *nicht* berücksichtigt werden, müssen beide Varianten als äquivalent erkannt werden, obwohl sie nicht durch Drehung zur Deckung gebracht werden können.

Bei der *Stereoisomerie* unterscheiden sich die Isomere in ihrer räumlichen Konformation, ein Vergleich der Strukturgraphen allein reicht nicht aus, um sie zu unterscheiden. Zusätzlich müssen weitere Merkmale wie Bindungswinkel, 3D Koordinaten, Spiegelebene und rotierbare Bindungen berechnet und in den Vergleich einbezogen werden. Beispiele dazu zeigt Tabelle C.3.

Die Unterscheidbarkeit von Stereoisomeren ist extrem wichtig. Zum Beispiel liegt die teratogene Wirkung von 2-(2,6-Dioxopiperidin-3-yl)isoindol-1,3-dion (Thalidomid), dem Wirkstoff in Contergan, hauptsächlich bei seinem S-Enantiomer [TCG93, S. 106]. Dies wurde bei der Herstellung von Contergan nicht erkannt und berücksichtigt. Allerdings hätte eine R-enantiomerenreine Formulierung die teratogene Wirkung auch nicht wesentlich vermindert, da sich die Enantiomere von Thalidomid in vivo ineinander umwandeln (racemisieren) bis wieder ein Gleichgewicht erreicht ist [EBR+95].

Ein weiteres Beispiel ist Methadon. Sein Enatiomer R-6-Dimethylamino-4,4-diphenylheptan-3-on ist ein starkes Analgetikum, während S-6-Dimethylamino-4,4-diphenylheptan-3-on stark antitussiv, aber kaum analgetisch wirkt.

Tabelle C.4: Spezielle Anforderungen an die Erkennung von Strukturgraphen - Tautomerie
Quelle: Eigene Darstellung

Typ	**Beispiel**	**Problem**
Keto-Enol-Tautomerie	 Ketoform 3-Methylbutan-2-on Enolform 3-Methylbut-2-en-2-ol	Soll Keto-Enol-Tautomerie bei einer Suche *nicht* berücksichtigt werden, müssen beide Varianten als äquivalent erkannt werden.
Amin-Imin-Tautomerie	 Amin Cyansäure Imin Isocyansäure	Soll Amin-Imin-Tautomerie bei einer Suche *nicht* berücksichtigt werden, müssen beide Varianten als äquivalent erkannt werden.
Lactam-Lactim-Tautomerie	 Lactam Pyrrolidin-2-on Lactim 3,4-Dihydro-2H-pyrrol-5-ol	Soll Lactam-Lactim-Tautomerie bei einer Suche *nicht* berücksichtigt werden, müssen beide Varianten als äquivalent erkannt werden.

Liegt *Tautomerie* (Definition 10) vor, ändern Atome oder Atomgruppen ihre Position im Strukturgraphen, zusätzlich können sich Bindungsordnungen ändern, wie in Tabelle C.4 gezeigt.

Definition 10. „Tautomerie *ist ein schneller, reversibler Übergang [durch Umlagerung von Atomen oder Atomgruppen; Anm. d. Verf.] von einer konstitutionsisomeren Form in eine andere. Die Gleichgewichtslage hängt von der Temperatur, dem Reaktionsmedium und der Energie beider Formen ab.“ [Sch04, S. 540]*

Falls diese Unterscheidung nicht gewünscht wird, müssen Isomere als äquivalent erkannt werden, auch wenn ihre Strukturgraphen verschieden sind. Der Grund dafür, die Unterscheidbarkeit nicht berücksichtigen zu wollen, liegt darin, dass Isomere in der Realität fast immer als teilweise sogar praktisch untrennbare Gemische vorkommen.

Wird ein solches Gemisch zum Beispiel nur mit seinem anteilig überwiegenden Isomer in einer Datenbank registriert, würde eine Suche mit dem anderen Isomer keine oder zu wenige Treffer liefern. Die Definition des Begriffs der chemischen Gleichheit von Strukturen hängt also stark vom Anwendungsfall ab.

D FPPC8 - Eine Modifikation des OpenBabel Fingerprints FP3 mit erweiterter Deskriptorkodierung

Bereits bei den ersten Versuchen der Optimierung der reduzierten Wörterbücher für den Wörterbuch-generierten OpenBabel Fingerprint vom Typ FP3 zeigte sich, dass auch bei guten Optimierungen nur wenige Bits des Fingerprints tatsächlich genutzt wurden. Daher wurde ein modifizierter exogener Fingerprint für OpenBabel entwickelt, welcher ein Byte dazu verwendet, neben der Existenz eines Merkmals auch die Anzahl der Treffer in der Zielstruktur zu kodieren. Es handelt sich also um structural keys mit erweiterter Deskriptorkodierung (vgl. Abschnitt III.1.6).

Den betroffenen Originalcode zeigt Listing D.1, die Modifikation Listing D.2.

Dieses Verfahren bietet neben einer höheren Selektivität den Vorteil, dass nicht alle möglichen bekannten Fälle blinder Stellen (vgl. Abschnitt III.2) explizit durch das Wörterbuch kodiert werden müssen. Für die Beispiele aus Tabelle III.3 und Abbildung III.4 reicht es, die abgebildeten Zwei-Ring-Systeme zu erfassen. Strukturen mit mehr als zwei Ringen werden dann automatisch durch die Anzahl der Treffer diskriminiert.

Die Wahl der Größe von einem Byte für einen Eintrag hat zwei Gründe. Erstens verfügen übliche Mikroprozessoren über spezielle Maschinenbefehle zur Manipulation

```
//Make fp size the smallest power of two to contain the patterns
unsigned int n=Getbitsperint();
while(n<smartsStrings.size())n*=2;
fp.resize(n/Getbitsperint());

for(n=0;n<smartsStrings.size();++n)
{
  OBSmartsPattern sp;
  sp.Init(smartsStrings[n]);
  if(sp.Match(*pmol,true))
  SetBit(fp, n);
}
```

Listing D.1: Originalcode zum Setzen der Bits in FP3

```
unsigned int o=0;
unsigned int m=0;
unsigned int i=0;
unsigned int n=0;

  if(!pmol)
    return false;

  //Read patterns file if it has not been done already

  if(smartsStrings.empty())
    ReadPatternFile(_patternsfile, smartsStrings);

  fp.resize(FPSIZE3);

  for(n=0;n<smartsStrings.size();++n)
    {
      OBSmartsPattern sp;
      sp.Init(smartsStrings[n]);

      if(sp.Match(*pmol)) {
          m=sp.GetUMapList().size();
          o=n*8;
          for(i=0;i<8;++i) {
                  if(i<m) {SetBit(fp, o+i);
             }
          }
      }
  }
```

Listing D.2: Code zum Setzen der Bits in FPPC8

von Bytes, so dass byteweise Operationen auf binären Fingerprints laufzeiteffizient implementiert werden können. Zweitens sind höhermolekulare ($n > 4$) kondensierte Ringsysteme vergleichsweise selten. Heptacen ($n = 7$) zum Beispiel konnte erst 2005 zweifelsfrei nachgewiesen [PPA05] und 2006 synthetisiert [MSN06] werden.

Mit einem Byte können, wie in Tabelle D.1 gezeigt, insgesamt neun Zustände kodiert werden. Ein Wörterbuch mit n Mustern kann also theoretisch 9^n verschiedene Fingerprints erzeugen. Für zum Beispiel $n = 64$ also $9^{64} = 1,179\,018\,458 \times 10^{61}$ Fingerprints.

Tabelle D.1: Byteweise binäre Trefferkodierung im FPPC8 Fingerprint
Quelle: Eigene Darstellung

Bitmuster	Anzahl der Treffer
00000000	$n = 0$
00000001	$n = 1$
00000011	$n = 2$
00000111	$n = 3$
00001111	$n = 4$
00011111	$n = 5$
00111111	$n = 6$
01111111	$n = 7$
11111111	$n \geq 8$

Die gewählte Kodierung ist daher ausreichend, um die Majorität der möglichen Anwendungsfälle zu erfassen.

E Mögliches Overtraining von structural keys am Beispiel des FPPC8 Algorithmus

Der in Anhang D vorgestellte FPPC8 Algorithmus verwendet SMARTS-Muster zur Spezifikation der Substruktursuchen, mittels welcher die Deskriptorbits gesetzt werden. SMARTS erlaubt es, über SMILES hinausgehende Muster zu definieren, wobei jedes gültige SMILES-Muster auch ein gültiges SMARTS-Muster ist. Hierbei besteht allerdings die Gefahr Muster zu definieren, welche Bedingung zwei aus Unterpunkt III.1.4 verletzen. Der resultierende Deskriptorenvektor ist zu selektiv (*overtrained*) und führt zu falschen Suchergebnissen.

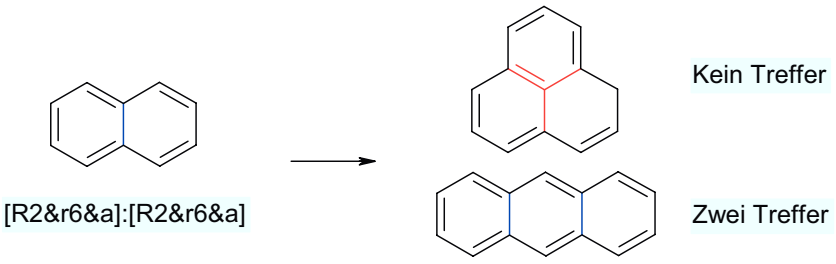


Abbildung E.1: Überselektives Muster [R2&r6&a]:[R2&r6&a]
Quelle: Eigene Darstellung

Bei dem Versuch, das in Tabelle III.3 gezeigte Problem des FP2 Algorithmus auszugleichen, wurde zunächst das SMARTS-Muster [R2&r6&a]:[R2&r6&a] verwendet. Dies führte dazu, dass 1H-Phenalen wie in Abbildung E.1 gezeigt fälschlicherweise unterschlagen wurde, da das zentrale Kohlenstoffatom *auch* mit drei Ringen verbunden

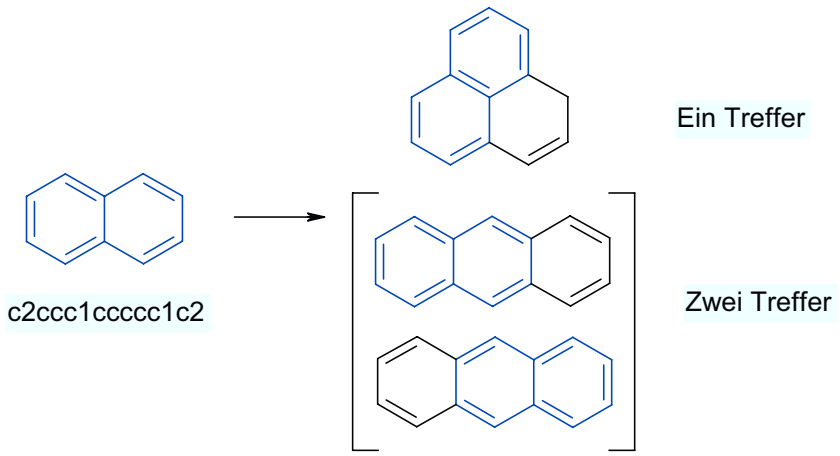


Abbildung E.2: Korrekte Lösung mit c2ccc1ccccc1c2
Quelle: Eigene Darstellung

ist. Das korrigierte SMARTS-Muster [R2,R3&r6&a]:[R2,R3&r6&a] lieferte dann das erwartete Ergebnis: Naphthalen ist eine Substruktur von 1H-Phenalen und Anthracen.

Letztendlich wurde das SMILES-Muster von Naphthalen, c2ccc1ccccc1c2, verwendet, da es ebenfalls, wie in Abbildung E.2 gezeigt, zu korrekten Ergebnissen führt und leichter verständlich ist.

F Der Aufbau von Pgchem::Tigress

F.1 Binäre Deskriptorenvektoren

Pgchem::Tigress verwendet den in Abbildung F.1 gezeigten zweiteiligen binären Deskriptorenvektor mit einer Gesamtlänge von 1 536 Bit.

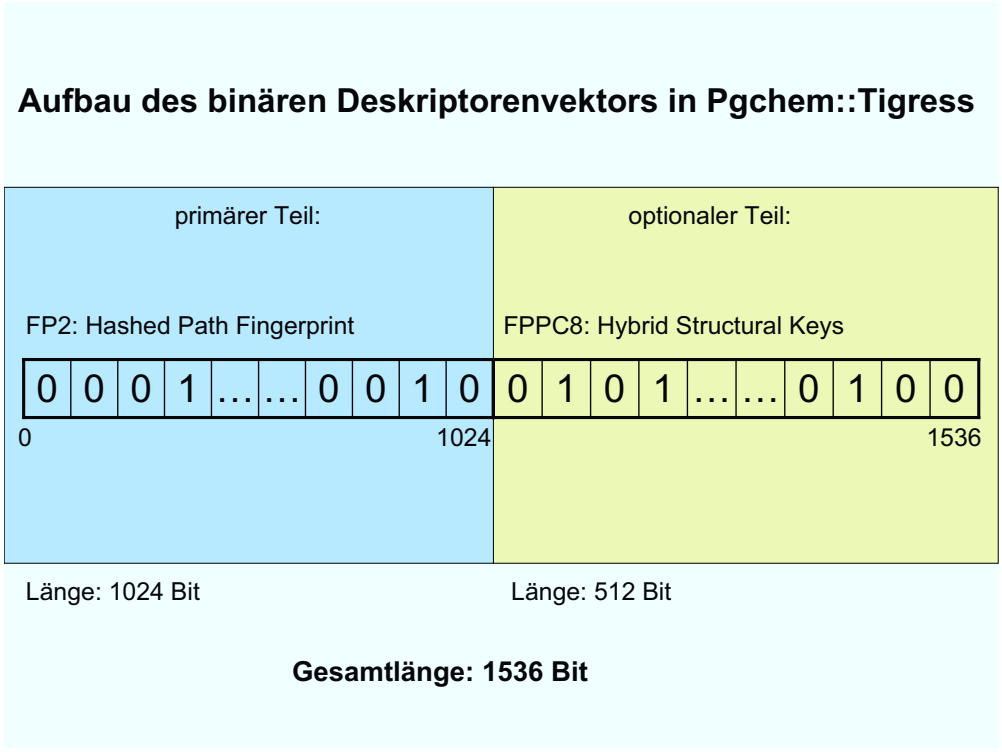


Abbildung F.1: Aufbau des binären Deskriptorenvektors in Pgchem:Tigress
Quelle: Eigene Darstellung

Der primäre Teil ist 1 024 Bit lang, enthält einen in Unterabschnitt III.1.7 beschriebenen hashed path fingerprint und wird durch den OpenBabel FP2 Algorithmus

generiert. Der optionale Teil ist 512 Bit lang und enthält structural keys mit hybrider Deskriptorkodierung, wie sie in Absatz III.1.6 beschrieben sind. Er wird durch den im Rahmen dieser Arbeit neu entwickelten OpenBabel FPPC8 Algorithmus generiert.

Tabelle F.1: Durchschnittliche Strukturgröße in verschiedenen Chemikalienkatalogen Quelle: Eigene Darstellung anhand von [Che09b] [May08b] [Asi08b] und [Asi08a]

Katalog	Jahr	Ø Strukturgröße (Byte)
ChemCollect Katalog	2009	1 574
Maybridge Screening Compounds	2008	1 617
Asinex Platinum Collection	2008	2 950
Asinex Gold Collection	2008	2 489

Die Länge von 1 024 Bit des primären Teils ist durch OpenBabel vorgegeben, die Länge des optionalen Teils ergibt sich aus dem Produkt der gewählten maximalen Wörterbuchgröße von 64 Mustern und einem Byte pro Muster. Mit einer Gesamtlänge von 192 Byte erfüllt dieser Fingerprint Anforderung vier aus Unterpunkt III.1.4, wie ein Vergleich mit den in Tabelle F.1 aufgeführten durchschnittlichen Strukturgrößen in verschiedenen Chemikalienkatalogen zeigt.

Der Aufbau folgt dem *fail fast* Prinzip gemäß Definition 11. Da der optionale Teil leer sein darf, wird zuerst der primäre Teil durchsucht.

Definition 11. *Das fail fast Prinzip besagt, dass eine Operation zum frühest möglichen Zeitpunkt abgebrochen wird, an dem ein Abbruchkriterium erfüllt ist: „. . .when a problem occurs, it fails immediately . . .“ [Jim04, S. 21]*

F.2 GiST

PostgreSQL als Basissystem für Pgchem::Tigress erlaubt es domänenspezifische Indextypen zu implementieren. Solche domänenspezifische Indextypen können entweder einen internen Mechanismus zur Verwaltung von B-Bäumen nutzen oder implementieren eine eigene Indexstruktur. PostgreSQL bietet dazu die Auswahl zwischen zwei Indexierungsmethoden, Generalized Search Tree (GiST) und Generalized Inverted Index (GIN). Pgchem::Tigress verwendet GiST.

Das GiST Konzept wurde 1995 vorgestellt [HNP95] und wird heute hauptsächlich vom GiST Indexing Project der University of California in Berkeley erforscht und

weiterentwickelt. GiST ist ein balancierter Baum bestehend aus <Schlüssel, Zeiger> Paaren und darauf definierte Methoden. Die Schlüssel dürfen beliebige Datentypen sein, die ein Merkmal kodieren, welches auch für alle Schlüssel, die über den zu dem Schlüssel gehörigen Zeiger erreicht werden können, erfüllt ist.

Der abstrakte Datentyp GiST (aus [HNP95], Abschnitt 3.1, 3.3 und 3.4.1 gekürzt übernommen und übersetzt)

Ein GiST stellt einen balancierten Baum mit variabler Verzweigung zwischen kM und M, $\frac{2}{M} \leq k \leq \frac{1}{2}$. Der Wurzelknoten ist die einzige Ausnahme mit einer Verzweigung zwischen 2 und M. Dabei ist k eine Konstante, der *minimale Füllfaktor* des Baumes. Blattknoten sind Paare der Form (p, ptr), wobei p ein als Suchschlüssel geeignetes Prädikat und ptr ein Zeiger auf ein Datenelement ist. Nicht-Blattknoten sind Paare der Form (p, ptr), wobei p ein als Suchschlüssel geeignetes Prädikat und ptr ein Zeiger auf einen anderen Knoten ist.

Die Funktionen, die für einen GiST Index implementiert werden müssen, sind:

compress(E) für ein Indexelement $E = (p, ptr)$ wird ein Eintrag (π, ptr) zurückgegeben, wobei π eine komprimierte Repräsentation von p ist

decompress(E) für ein Indexelement $E = (\pi, ptr)$ mit π = compress(p) wird ein Eintrag $E = (r, ptr)$ zurückgegeben, so dass gilt: $p \rightarrow r$. $p \leftrightarrow r$ wird *nicht* vorausgesetzt, so dass bei dieser Operation Information verloren gehen kann

consistent(E, q) für ein Indexelement $E = (p, ptr)$ und ein Abfrageprädikat q; falls $p \wedge q$ garantiert nicht erfüllt werden kann, wird FALSCH zurückgegeben, ansonsten WAHR; anzumerken ist, dass ein Test auf Unerfüllbarkeit von $p \wedge q$ der fälschlicherweise WAHR zurückliefert die Korrektheit des Index nicht beeinträchtigt, sondern nur die Suchperformance einschränkt, weil unter Umständen irrelevante Teilbäume durchsucht werden müssen; die einfachste Implementierung von consistent(E, q) gibt also immer WAHR zurück

union(P) für eine Menge P von Indexelementen $(p_1, ptr_1), \ldots (p_n, ptr_n)$ wird ein Prädikat r zurückgegeben, das für alle Einträge $(p_1, ptr_1), \ldots (p_n, ptr_n)$ WAHR ist: $(p_1 \vee \ldots \vee p_n) \rightarrow r)$

Tabelle F.2: Schematischer Ablauf einer Suche in einem GiST Baum
Quelle: Eigene Darstellung anhand von [HNP95], Abschnitt 3.4.1

search(R,q)	
Eingabe	GiST Wurzelknoten R, Prädikat q.
Ausgabe	Alle Tupel für die q=WAHR.
Ablauf	Rekursiver Abstieg durch alle Pfade im Baum, deren Schlüssel zu q konsistent sind.
Suche in Teilbäumen	Falls R kein Blattknoten ist, überprüfe alle Einträge E in R auf consistent(E,q)=WAHR. Für alle passenden E suche weiter im Teilbaum, dessen Wurzelknoten durch $E.ptr$ referenziert wird.
Suche in Blattknoten	Falls R ein Blattknoten ist, überprüfe alle Einträge E in R auf consistent(E,q)=WAHR. Für alle passenden E überprüfe $E.ptr$ genauer gegen q oder liefere $E.ptr$ an den aufrufenden Prozess zurück, so dass dieser eine genaue Überprüfung vornehmen kann.

penalty(E_1, E_2) für zwei Indexelemente $E_1 = (p_1, ptr_1)$, $E_2(p_2, ptr_2)$ wird ein domänenabhängiges Strafmaß für das Einfügen von E_2 in den Teilbaum unterhalb von E_1 zurückgegeben; dies unterstützt das GiST Subsystem bei der Entscheidung, wo im Baum neue Elemente eingefügt werden sollen

picksplit(P) für eine Menge von Indexelementen P der Mächtigkeit $M + 1$, wird P in zwei Teilmengen P_1,P_2 geteilt, wobei jede mindestens die Mächtigkeit kM haben muss; die einfachste Implementierung von picksplit() teilt P in zwei möglichst gleich große Mengen, üblicherweise versucht man aber die Elemente so zu verteilen, dass die Wahrscheinlichkeit, zwei Teilbäume durchsuchen zu müssen, minimiert wird

Eine Suche läuft dann nach dem in Tabelle F.2 skizzierten Algorithmus ab.

Die GiST Implementierung für binäre Deskriptorenvektoren Die GiST Application Programming Interface (API) in PostgreSQL erfordert nur die Implementierung des Schlüsseldatentyps für die <Schlüssel, Zeiger> Paare, den Baum verwaltet das

```
typedef struct
{
  uint32 fp[FPSIZE];
} MOLFP;
```

Listing F.1: Der Datentyp MOLFP

```
typedef struct
{
  int4 len;
  int4 sizemf;
  int4 sizesmi;
  int4 disconnected;
  uint32 fp[FPSIZE];
  char inchikey[INCHIKEYSZ];
  char data[1];
} MOLECULE;
```

Listing F.2: Der Datentyp MOLECULE

GiST-Subsystem selber. Der interne Schlüsseldatentyp für molekulare Deskriptorenvektoren ist vom Typ MOLFP wie in Listing F.1 gezeigt.

Das Array fp enthält dabei den binären Deskriptorenvektor. FPSIZE ist eine zur Übersetzungszeit bekannte Konstante für die Größe des Arrays.

Allerdings schickt die Datenbank teilweise auch Elemente vom Typ MOLECULE (Listing F.2) an das GiST Subsystem, so dass einige Funktionen von MOLECULE nach MOLFP konvertieren müssen.

Die GiST Funktionen für binäre Deskriptorenvektoren in PGCHEM::TIGRESS sind wie folgt implementiert:

compress(E) für ein Indexelement $E = (MOLECULE, ptr)$ wird ein Eintrag $(MOLFP, ptr)$ zurückgegeben

decompress(E) für ein Indexelement $E = (MOLFP, ptr)$ wird ein Eintrag $E = (MOLFP, ptr)$ zurückgegeben, da es nicht möglich ist, den Strukturgraphen aus seinem Deskriptorenvektor zu rekonstruieren

consistent(E, q) gibt WAHR zurück, falls der Deskriptorenvektor q im Deskriptorenvektor von E enthalten ist (bitweises UND von q und E und anschließendes bitweises XOR des Zwischenergebnisses mit q), sonst FALSCH

same(E, q) gibt WAHR zurück, falls der Deskriptorenvektor q gleich dem Deskriptorenvektor von E ist (Vergleich von E mit q), sonst FALSCH; dies ist eine durch PostgreSQL zusätzlich geforderte Funktion

union(P) gibt das Ergebnis des bitweisen ODER aller MOLFP Deskriptorenvektoren in P zurück

penalty(E_1, E_2) gibt die Soergel Distanz D_S (vgl. Anhang A) zwischen E_1 und E_2 zurück

picksplit(P) erzeugt zwei neue Indexseiten P_1 und P_2 aus der Indexseite P und gibt diese zurück; die Aufteilung erfolgt mittels einem modifizierten Quadratic Split Algorithmus nach Guttman (vgl. [Gut84]), wobei die interne Distanzberechnung analog zur penalty() Funktion D_S benutzt

G Beispiellösung eines relaxierten binären LP mit dem Computer Algebra System Maxima

Lineares Programm mit 198 Strukturvariablen und 397 Restriktionen:

```
(%i604) load("simplex")$
u1:8.00$u2:8.00$u3:8.00$u4:7.99$u5:7.98$u6:7.93$u7:7.82$u8:7.82$u9:6.90$
u10:6.90$u11:6.90$u12:6.88$u13:6.88$u14:6.88$u15:6.84$u16:6.84$u17:6.79$
u18:6.74$u19:6.74$u20:6.74$u21:6.72$u22:6.63$u23:6.50$u24:6.50$u25:6.50$
u26:6.40$u27:6.40$u28:6.40$u29:5.98$u30:5.71$u31:5.28$u32:5.12$u33:5.12$
u34:5.12$u35:4.95$u36:4.95$u37:4.95$u38:4.29$u39:4.02$u40:4.02$u41:3.63$
u42:3.63$u43:3.63$u44:3.63$u45:3.62$u46:3.62$u47:3.62$u48:3.17$u49:3.11$
u50:3.11$u51:3.11$u52:3.08$u53:2.88$u54:2.88$u55:2.88$u56:2.82$u57:2.74$
u58:2.57$u59:2.57$u60:2.57$u61:2.57$u62:2.39$u63:2.35$u64:2.35$u65:2.35$
u66:2.22$u67:1.99$u68:1.99$u69:1.96$u70:1.96$u71:1.85$u72:1.83$u73:1.73$
u74:1.50$u75:1.37$u76:1.37$u77:1.37$u78:1.36$u79:1.33$u80:1.33$u81:1.33$
u82:1.33$u83:1.32$u84:1.32$u85:1.32$u86:1.11$u87:1.06$u88:1.03$u89:1.03$
u90:1.03$u91:1.01$u92:0.99$u93:0.97$u94:0.90$u95:0.82$u96:0.77$u97:0.77$
u98:0.77$u99:0.67$u100:0.65$u101:0.65$u102:0.64$u103:0.64$u104:0.64$
u105:0.64$u106:0.53$u107:0.53$u108:0.53$u109:0.50$u110:0.50$u111:0.50$
u112:0.49$u113:0.49$u114:0.49$u115:0.49$u116:0.49$u117:0.49$u118:0.49$
u119:0.43$u120:0.37$u121:0.37$u122:0.37$u123:0.34$u124:0.34$u125:0.32$
u126:0.32$u127:0.32$u128:0.27$u129:0.25$u130:0.25$u131:0.25$u132:0.21$
u133:0.21$u134:0.21$u135:0.15$u136:0.15$u137:0.15$u138:0.15$u139:0.15$
u140:0.15$u141:0.13$u142:0.13$u143:0.13$u144:0.13$u145:0.12$u146:0.12$
u147:0.12$u148:0.12$u149:0.12$u150:0.12$u151:0.12$u152:0.12$u153:0.12$
u154:0.12$u155:0.12$u156:0.12$u157:0.12$u158:0.12$u159:0.12$u160:0.11$
u161:0.11$u162:0.11$u163:0.08$u164:0.08$u165:0.08$u166:0.08$u167:0.08$
u168:0.08$u169:0.08$u170:0.08$u171:0.08$u172:0.08$u173:0.08$u174:0.06$
u175:0.06$u176:0.06$u177:0.06$u178:0.04$u179:0.04$u180:0.02$u181:0.02$
```

```
u182:0.02$u183:0.02$u184:0.02$u185:0.02$u186:0.02$u187:0.02$u188:0.02$
u189:0.02$u190:0.02$u191:0.02$u192:0.02$u193:0.01$u194:0.01$u195:0.01$
u196:0.01$u197:0.01$u198:0.01$G:64$
minimize_lp(u1*x1+u2*x2+u3*x3+u4*x4+u5*x5+u6*x6+u7*x7+u8*x8+u9*x9+
u10*x10+u11*x11+u12*x12+u13*x13+u14*x14+u15*x15+u16*x16+u17*x17+
u18*x18+u19*x19+u20*x20+u21*x21+u22*x22+u23*x23+u24*x24+u25*x25+
u26*x26+u27*x27+u28*x28+u29*x29+u30*x30+u31*x31+u32*x32+u33*x33+
u34*x34+u35*x35+u36*x36+u37*x37+u38*x38+u39*x39+u40*x40+u41*x41+
u42*x42+u43*x43+u44*x44+u45*x45+u46*x46+u47*x47+u48*x48+u49*x49+
u50*x50+u51*x51+u52*x52+u53*x53+u54*x54+u55*x55+u56*x56+u57*x57+
u58*x58+u59*x59+u60*x60+u61*x61+u62*x62+u63*x63+u64*x64+u65*x65+
u66*x66+u67*x67+u68*x68+u69*x69+u70*x70+u71*x71+u72*x72+u73*x73+
u74*x74+u75*x75+u76*x76+u77*x77+u78*x78+u79*x79+u80*x80+u81*x81+
u82*x82+u83*x83+u84*x84+u85*x85+u86*x86+u87*x87+u88*x88+u89*x89+
u90*x90+u91*x91+u92*x92+u93*x93+u94*x94+u95*x95+u96*x96+u97*x97+
u98*x98+u99*x99+u100*x100+u101*x101+u102*x102+u103*x103+u104*x104+
u105*x105+u106*x106+u107*x107+u108*x108+u109*x109+u110*x110+
u111*x111+u112*x112+u113*x113+u114*x114+u115*x115+u116*x116+
u117*x117+u118*x118+u119*x119+u120*x120+u121*x121+u122*x122+
u123*x123+u124*x124+u125*x125+u126*x126+u127*x127+u128*x128+
u129*x129+u130*x130+u131*x131+u132*x132+u133*x133+u134*x134+
u135*x135+u136*x136+u137*x137+u138*x138+u139*x139+u140*x140+
u141*x141+u142*x142+u143*x143+u144*x144+u145*x145+u146*x146+
u147*x147+u148*x148+u149*x149+u150*x150+u151*x151+u152*x152+
u153*x153+u154*x154+u155*x155+u156*x156+u157*x157+u158*x158+
u159*x159+u160*x160+u161*x161+u162*x162+u163*x163+u164*x164+
u165*x165+u166*x166+u167*x167+u168*x168+u169*x169+u170*x170+
u171*x171+u172*x172+u173*x173+u174*x174+u175*x175+u176*x176+
u177*x177+u178*x178+u179*x179+u180*x180+u181*x181+u182*x182+
u183*x183+u184*x184+u185*x185+u186*x186+u187*x187+u188*x188+
u189*x189+u190*x190+u191*x191+u192*x192+u193*x193+u194*x194+
u195*x195+u196*x196+u197*x197+u198*x198,
[x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12+x13+x14+x15+x16+x17+
x18+x19+x20+x21+x22+x23+x24+x25+x26+x27+x28+x29+x30+x31+x32+
x33+x34+x35+x36+x37+x38+x39+x40+x41+x42+x43+x44+x45+x46+x47+
```

```
x48+x49+x50+x51+x52+x53+x54+x55+x56+x57+x58+x59+x60+x61+x62+
x63+x64+x65+x66+x67+x68+x69+x70+x71+x72+x73+x74+x75+x76+x77+
x78+x79+x80+x81+x82+x83+x84+x85+x86+x87+x88+x89+x90+x91+x92+
x93+x94+x95+x96+x97+x98+x99+x100+x101+x102+x103+x104+x105+x106+
x107+x108+x109+x110+x111+x112+x113+x114+x115+x116+x117+x118+
x119+x120+x121+x122+x123+x124+x125+x126+x127+x128+x129+x130+
x131+x132+x133+x134+x135+x136+x137+x138+x139+x140+x141+x142+
x143+x144+x145+x146+x147+x148+x149+x150+x151+x152+x153+x154+
x155+x156+x157+x158+x159+x160+x161+x162+x163+x164+x165+x166+
x167+x168+x169+x170+x171+x172+x173+x174+x175+x176+x177+x178+
x179+x180+x181+x182+x183+x184+x185+x186+x187+x188+x189+x190+
x191+x192+x193+x194+x195+x196+x197+x198>=G,
x1>=0, x1<=1, x2>=0, x2<=1, x3>=0, x3<=1, x4>=0, x4<=1, x5>=0,
x5<=1, x6>=0, x6<=1, x7>=0, x7<=1, x8>=0, x8<=1, x9>=0, x9<=1,
x10>=0, x10<=1, x11>=0, x11<=1, x12>=0, x12<=1, x13>=0,
x13<=1, x14>=0, x14<=1, x15>=0, x15<=1, x16>=0, x16<=1, x17>=0,
x17<=1, x18>=0, x18<=1, x19>=0, x19<=1, x20>=0, x20<=1, x21>=0,
x21<=1, x22>=0, x22<=1, x23>=0, x23<=1, x24>=0, x24<=1, x25>=0,
x25<=1, x26>=0, x26<=1, x27>=0, x27<=1, x28>=0, x28<=1, x29>=0,
x29<=1, x30>=0, x30<=1, x31>=0, x31<=1, x32>=0, x32<=1, x33>=0,
x33<=1, x34>=0, x34<=1, x35>=0, x35<=1, x36>=0, x36<=1, x37>=0,
x37<=1, x38>=0, x38<=1, x39>=0, x39<=1, x40>=0, x40<=1, x41>=0,
x41<=1, x42>=0, x42<=1, x43>=0, x43<=1, x44>=0, x44<=1, x45>=0,
x45<=1, x46>=0, x46<=1, x47>=0, x47<=1, x48>=0, x48<=1, x49>=0,
x49<=1, x50>=0, x50<=1, x51>=0, x51<=1, x52>=0, x52<=1, x53>=0,
x53<=1, x54>=0, x54<=1, x55>=0, x55<=1, x56>=0, x56<=1, x57>=0,
x57<=1, x58>=0, x58<=1, x59>=0, x59<=1, x60>=0, x60<=1, x61>=0,
x61<=1, x62>=0, x62<=1, x63>=0, x63<=1, x64>=0, x64<=1, x65>=0,
x65<=1, x66>=0, x66<=1, x67>=0, x67<=1, x68>=0, x68<=1, x69>=0,
x69<=1, x70>=0, x70<=1, x71>=0, x71<=1, x72>=0, x72<=1, x73>=0,
x73<=1, x74>=0, x74<=1, x75>=0, x75<=1, x76>=0, x76<=1, x77>=0,
x77<=1, x78>=0, x78<=1, x79>=0, x79<=1, x80>=0, x80<=1, x81>=0,
x81<=1, x82>=0, x82<=1, x83>=0, x83<=1, x84>=0, x84<=1, x85>=0,
x85<=1, x86>=0, x86<=1, x87>=0, x87<=1, x88>=0, x88<=1, x89>=0,
x89<=1, x90>=0, x90<=1, x91>=0, x91<=1, x92>=0, x92<=1, x93>=0,
```

```
x93<=1, x94>=0, x94<=1, x95>=0, x95<=1, x96>=0, x96<=1, x97>=0,
x97<=1, x98>=0, x98<=1, x99>=0, x99<=1, x100>=0, x100<=1, x101>=0,
x101<=1, x102>=0, x102<=1, x103>=0, x103<=1, x104>=0, x104<=1,
x105>=0, x105<=1, x106>=0, x106<=1, x107>=0, x107<=1, x108>=0,
x108<=1, x109>=0, x109<=1, x110>=0, x110<=1, x111>=0, x111<=1,
x112>=0, x112<=1, x113>=0, x113<=1, x114>=0, x114<=1, x115>=0,
x115<=1, x116>=0, x116<=1, x117>=0, x117<=1, x118>=0, x118<=1,
x119>=0, x119<=1, x120>=0, x120<=1, x121>=0, x121<=1, x122>=0,
x122<=1, x123>=0, x123<=1, x124>=0, x124<=1, x125>=0, x125<=1,
x126>=0, x126<=1, x127>=0, x127<=1, x128>=0, x128<=1, x129>=0,
x129<=1, x130>=0, x130<=1, x131>=0, x131<=1, x132>=0, x132<=1,
x133>=0, x133<=1, x134>=0, x134<=1, x135>=0, x135<=1, x136>=0,
x136<=1, x137>=0, x137<=1, x138>=0, x138<=1, x139>=0, x139<=1,
x140>=0, x140<=1, x141>=0, x141<=1, x142>=0, x142<=1, x143>=0,
x143<=1, x144>=0, x144<=1, x145>=0, x145<=1, x146>=0, x146<=1,
x147>=0, x147<=1, x148>=0, x148<=1, x149>=0, x149<=1, x150>=0,
x150<=1, x151>=0, x151<=1, x152>=0, x152<=1, x153>=0, x153<=1,
x154>=0, x154<=1, x155>=0, x155<=1, x156>=0, x156<=1, x157>=0,
x157<=1, x158>=0, x158<=1, x159>=0, x159<=1, x160>=0, x160<=1,
x161>=0, x161<=1, x162>=0, x162<=1, x163>=0, x163<=1, x164>=0,
x164<=1, x165>=0, x165<=1, x166>=0, x166<=1, x167>=0, x167<=1,
x168>=0, x168<=1, x169>=0, x169<=1, x170>=0, x170<=1, x171>=0,
x171<=1, x172>=0, x172<=1, x173>=0, x173<=1, x174>=0, x174<=1,
x175>=0, x175<=1, x176>=0, x176<=1, x177>=0, x177<=1, x178>=0,
x178<=1, x179>=0, x179<=1, x180>=0, x180<=1, x181>=0, x181<=1,
x182>=0, x182<=1, x183>=0, x183<=1, x184>=0, x184<=1, x185>=0,
x185<=1, x186>=0, x186<=1, x187>=0, x187<=1, x188>=0, x188<=1,
x189>=0, x189<=1, x190>=0, x190<=1, x191>=0, x191<=1, x192>=0,
x192<=1, x193>=0, x193<=1, x194>=0, x194<=1, x195>=0, x195<=1,
x196>=0, x196<=1, x197>=0, x197<=1, x198>=0, x198<=1]),
nonnegative_lp=true;

(%o804) [5.069999999999353,[x99=0,x98=0,x97=0,
x96=0,x95=0,x94=0,
x93=0,x92=0,x91=0,x90=0,x9=0,x89=0,x88=0,x87=0,
```

```
x86=0,x85=0,x84=0,x83=0,x82=0,x81=0,x80=0,x8=0,
x79=0,x78=0,x77=0,x76=0,x75=0,x74=0,x73=0,x72=0,
x71=0,x70=0,
x7=0,x69=0,x68=0,x67=0,x66=0,x65=0,x64=0,x63=0,
x62=0,x61=0,x60=0,x6=0,x59=0,x58=0,x57=0,x56=0,
x55=0,x54=0,
x53=0,x52=0,x51=0,x50=0,x5=0,x49=0,x48=0,x47=0,
x46=0,x45=0,x44=0,x43=0,x42=0,x41=0,x40=0,x4=0,
x39=0,x38=0,
x37=0,x36=0,x35=0,x34=0,x33=0,x32=0,x31=0,x30=0,
x3=0,x29=0,x28=0,x27=0,x26=0,x25=0,x24=0,x23=0,
x22=0,x21=0,
x20=0,x2=0,x198=1,x197=1,x196=1,x195=1,x194=1,
x193=1,x192=1,x191=1,x190=1,x19=0,x189=1,x188=1,
x187=1,x186=1,
x185=1,x184=1,x183=1,x182=1,x181=1,x180=1,x18=0,
x179=1,x178=1,x177=1,x176=1,x175=1,x174=1,x173=1,
x172=1,x171=1,
x170=1,x17=0,x169=1,x168=1,x167=1,x166=1,x165=1,
x164=1,x163=1,x162=1,x161=1,x160=1,x16=0,x159=1,
x158=1,x157=1,
x156=1,x155=1,x154=1,x153=1,x152=1,x151=1,x150=1,
x15=0,x149=1,x148=1,x147=1,x146=1,x145=1,x144=1,
x143=1,x142=1,
x141=1,x140=1,x14=0,x139=1,x138=1,x137=1,x136=1,
x135=1,x134=0,x133=0,x132=0,x131=0,x130=0,x13=0,
x129=0,x128=0,
x127=0,x126=0,x125=0,x124=0,x123=0,x122=0,x121=0,
x120=0,x12=0,x119=0,x118=0,x117=0,x116=0,x115=0,
x114=0,x113=0,
x112=0,x111=0,x110=0,x11=0,x109=0,x108=0,x107=0,
x106=0,x105=0,x104=0,x103=0,x102=0,x101=0,x100=0,
x10=0,x1=0]]
(%i807) time(\%o804)
(%o807) [34.92]
```

H ANOVA Protokolle

Response 1 **Delta Selectivity**

ANOVA for selected factorial model

Analysis of variance table [Partial sum of squares - Type III]

Source	Sum of Squares	df	Mean Square	F Value	p-value Prob > F	
Model	0.20	2	0.099	45.45	0.1043	not significant
A-Chromosome size	*0.090*	*1*	*0.090*	*41.17*	*0.0984*	
B-Evolutions	*0.11*	*1*	*0.11*	*49.73*	*0.0897*	
Curvature	0.044	1	0.044	19.96	0.1402	not significant
Residual	2.180E-003	1	2.180E-003			
Cor Total	0.24	4				

The "Model F-value" of 45.45 implies the model is not significant relative to the noise. There is a 10.43 % chance that a "Model F-value" this large could occur due to noise.

Values of "Prob > F" less than 0.0500 indicate model terms are significant.
In this case there are no significant model terms.
Values greater than 0.1000 indicate the model terms are not significant.
If there are many insignificant model terms (not counting those required to support hierarchy), model reduction may improve your model.

The "Curvature F-value" of 19.96 implies the curvature (as measured by difference between the average of the center points and the average of the factorial points) in the design space is not significant relative to the noise. There is a 14.02% chance that a "Curvature F-value" this large could occur due to noise.

Std. Dev.	0.047	R-Squared	0.9891
Mean	0.44	Adj R-Squared	0.9674
C.V. %	10.52	Pred R-Squared	N/A
PRESS	N/A	Adeq Precision	15.059

Case(s) with leverage of 1.0000: Pred R-Squared and PRESS statistic not defined

"Adeq Precision" measures the signal to noise ratio. A ratio greater than 4 is desirable. Your ratio of 15.059 indicates an adequate signal. This model can be used to navigate the design space.

Factor	Coefficient Estimate	df	Standard Error	95% CI Low	95% CI High	VIF
Intercept	0.40	1	0.023	0.10	0.69	
A-Chromosome size	0.15	1	0.023	-0.15	0.45	1.00
B-Evolutions	0.16	1	0.023	-0.13	0.46	1.00
Center Point	0.23	1	0.052	-0.43	0.90	1.00

Final Equation in Terms of Coded Factors:

Fitness =
+0.40
+0.15 * A
+0.16 * B

Final Equation in Terms of Actual Factors:

Fitness =
+0.031440
+5.35019E-003 * Chromosome size
+8.44308E-003 * Evolutions

Abbildung H.1: ANOVA der Selektivitätsveränderung für den Sampling Algorithmus
Quelle: Eigene Darstellung mit Design-Ease®

Response 2 **Running Time**

ANOVA for selected factorial model

Analysis of variance table [Partial sum of squares - Type III]

Source	Sum of Squares	df	Mean Square	F Value	p-value Prob > F	
Model	2.686E+011	1	2.686E+011	8.69	0.0984	not significant
B-Evolutions	*2.686E+011*	*1*	*0.11*	*8.69*	*0.0984*	
Curvature	2.140E+009	1	2.140E+009	0.069	0.8170	not significant
Residual	6.181E+010	2	3.090E+010			
Cor Total	3.326E+011	4				

The "Model F-value" of 8.69 implies there is a 9.84 % chance that a "Model F-value" this large could occur due to noise.

Values of "Prob > F" less than 0.0500 indicate model terms are significant.
In this case there are no significant model terms.
Values greater than 0.1000 indicate the model terms are not significant.
If there are many insignificant model terms (not counting those required to support hierarchy), model reduction may improve your model.

The "Curvature F-value" of 0.07 implies the curvature (as measured by difference between the average of the center points and the average of the factorial points) in the design space is not significant relative to the noise. There is a 81.70% chance that a "Curvature F-value" this large could occur due to noise.

Std. Dev.	1.758E+005	R-Squared	0.8130
Mean	3.876E+005	Adj R-Squared	0.7194
C.V. %	45.35	Pred R-Squared	N/A
PRESS	N/A	Adeq Precision	3.806

Case(s) with leverage of 1.0000: Pred R-Squared and PRESS statistic not defined

"Adeq Precision" measures the signal to noise ratio. A ratio of 3.81 indicates an inadequate signal and we should not use this model to navigate the design space.

Factor	Coefficient Estimate	df	Standard Error	95% CI Low	95% CI High	VIF
Intercept	3.980E+005	1	87896.42	19786.50	7.762E+005	
B-Evolutions	2.592E+005	1	87896.42	-1.190E+005	6.373E+005	1.00
Center Point	-51726.25	1	1.965E+005	-8.974E+005	7.939E+005	1.00

Final Equation in Terms of Coded Factors:

Running time =
+3.980E+005
+2.592E+005 * B

Final Equation in Terms of Actual Factors:

Running time =
+1.25530E+005
+13289.98718 * Evolutions

Abbildung H.2: ANOVA der Laufzeitveränderung für den Sampling Algorithmus
Quelle: Eigene Darstellung mit Design-Ease®

Response 1 **Delta Selectivity**

ANOVA for selected factorial model

Analysis of variance table [Partial sum of squares - Type III]

Source	Sum of Squares	df	Mean Square	F Value	p-value Prob > F	
Model	0.49	3	0.16	18.47	0.0083	significant
A-Population size	*0.16*	*1*	*0.16*	*18.29*	*0.0129*	
B-Evolutions	*0.088*	*1*	*0.088*	*9.95*	*0.0344*	
C-Chromosome Size	*0.24*	*1*	*0.24*	*27.18*	*0.0065*	
Curvature	0.049	1	0.049	5.49	0.0792	not significant
Residual	0.035	4	8.843E-003			
Cor Total	0.24	8				

The "Model F-value" of 18.47 implies the model is significant. There is only a 0.83 % chance that a "Model F-value" this large could occur due to noise.

Values of "Prob > F" less than 0.0500 indicate model terms are significant.
In this case A,B,C are significant model terms.
Values greater than 0.1000 indicate the model terms are not significant.
If there are many insignificant model terms (not counting those required to support hierarchy), model reduction may improve your model.

The "Curvature F-value" of 5.49 implies there is curvature (as measured by difference between the average of the center points and the average of the factorial points) in the design space.
There is only a 7.92% chance that a "Curvature F-value" this large could occur due to noise.

Std. Dev.	0.094	R-Squared	0.9327
Mean	0.55	Adj R-Squared	0.8822
C.V. %	17.17	Pred R-Squared	N/A
PRESS	N/A	Adeq Precision	11.996

Case(s) with leverage of 1.0000: Pred R-Squared and PRESS statistic not defined

"Adeq Precision" measures the signal to noise ratio. A ratio greater than 4 is desirable. Your ratio of 15.059 indicates an adequate signal. This model can be used to navigate the design space.

Factor	Coefficient Estimate	df	Standard Error	95% CI Low	95% CI High	VIF
Intercept	0.52	1	0.033	0.43	0.61	
A-Population size	0.14	1	0.033	0.050	0.23	1.00
B-Evolutions	0.10	1	0.033	0.013	0.20	1.00
C-Chromosome size	0.17		0.033	0.081	0.27	1.00
Center Point	0.23	1	0.1	-0.043	0.51	1.00

Final Equation in Terms of Coded Factors:

Delta Selectivity =
+0.52
+0.14 * A
+0.10 * B
+0.17 * C

Final Equation in Terms of Actual Factors:

Delta Selectivity =
+0.046773
+2.87284E-003 * Population Size
+2.11871E-003 * Evolutions
+6.19051E-003 * Chromosome size

Abbildung H.3: ANOVA der Selektivitätsveränderung für den genetischen Algorithmus
Quelle: Eigene Darstellung mit Design-Ease®

Response 2 **Running Time**

Transform: Base 10 log **Constant: 0**

ANOVA for selected factorial model

Analysis of variance table [Partial sum of squares - Type III]

Source	Sum of Squares	df	Mean Square	F Value	p-value Prob > F	
Model	13.77	2	6.89	553.95	< 0.0001	significant
A-Population size	*8.72*	*1*	*8.72*	*701.52*	< 0.0001	
B-Evolutions	*5.05*	*1*	*5.05*	*406.39*	< 0.0001	
Curvature	0.049	1	1.48	118.90	0.0001	significant
Residual	0.062	5	0.012			
Cor Total	15.31	8				

The "Model F-value" of 553.95 implies the model is significant. There is only a 0.01 % chance that a "Model F-value" this large could occur due to noise.

Values of "Prob > F" less than 0.0500 indicate model terms are significant.
In this case A,B are significant model terms.
Values greater than 0.1000 indicate the model terms are not significant.
If there are many insignificant model terms (not counting those required to support hierarchy), model reduction may improve your model.

The "Curvature F-value" of 118.90 implies there is curvature (as measured by difference between the average of the center points and the average of the factorial points) in the design space.
There is only a 0.01% chance that a "Curvature F-value" this large could occur due to noise.

Std. Dev.	0.11	R-Squared	0.9955
Mean	6.17	Adj R-Squared	0.9937
C.V. %	1.81	Pred R-Squared	N/A
PRESS	N/A	Adeq Precision	49.475

Case(s) with leverage of 1.0000: Pred R-Squared and PRESS statistic not defined

"Adeq Precision" measures the signal to noise ratio. A ratio greater than 4 is desirable. Your ratio of 49.475 indicates an adequate signal. This model can be used to navigate the design space.

Factor	Coefficient Estimate	df	Standard Error	95% CI Low	95% CI High	VIF
Intercept	6.03	1	0.039	5.93	6.13	
A-Population size	1.04	1	0.039	0.94	1.15	1.00
B-Evolutions	0.79	1	0.039	0.69	0.90	1.00
Center Point	1.29	1	0.12	0.99	1.59	1.00

Final Equation in Terms of Coded Factors:

Log_{10} (Running Time) =
+6.03
+1.04 * A
+0.79 * B

Final Equation in Terms of Actual Factors:

Log_{10} (Running Time) =
+4.15071
+0.021093 * Population Size
+0.016054 * Evolutions

Abbildung H.4: ANOVA der Laufzeitveränderung für den genetischen Algorithmus
Quelle: Eigene Darstellung mit Design-Ease®

I ACS Suche

http://pubs.acs.org/action/doSearch?searchText=&action=search&type=within&prevSearch=%255Btitle%253A%2Bscreening%2BOR%2Bclustering%2BOR%2Bfingerprint%2BOR%2Bsimilarity%2BOR%2Bsubstructure%2BOR%2Bdatabase%2BOR%2Bindex%2BOR%2Bscore%255D&target=&targetTab=research&filter=&startPage=&func=showSearch&result=true&stemming=&sortBy=date&pageSize=20&restrict=&displaySummary=&startYear=2000&startMonth=&endYear=2009&endMonth=&pubDateRange=coverDateRange&title=screening+OR+clustering+OR+fingerprint+OR+similarity+OR+substructure+OR+database+OR+index+OR+score&publication=40026030&saveSearchName=&alertme=Never&articleType=primary_article

J Das verwendete Basiswörterbuch

[#2]
[#3]
[#4]
[#5]
[#6]
[#7]
[#8]
[#9]
[#10]
[#11]
[#12]
[#13]
[#14]
[#15]
[#16]
[#17]
[#18]
[#19]
[#20]
[#21]
[#22]
[#23]
[#24]
[#25]
[#26]
[#27]
[#28]
[#29]
[#30]
[#31]
[#32]
[#33]
[#34]
[#35]
[#36]
[#37]
[#38]
[#39]
[#40]
[#41]
[#42]
[#43]
[#44]
[#45]
[#46]
[#47]
[#48]
[#49]
[#50]
[#51]
[#52]
[#53]
[#54]
[#55]
[#56]
[#57]
[#58]
[#59]
[#60]

```
[#61]
[#62]
[#63]
[#64]
[#65]
[#66]
[#67]
[#68]
[#69]
[#70]
[#71]
[#72]
[#73]
[#74]
[#75]
[#76]
[#77]
[#78]
[#79]
[#80]
[#81]
[#82]
[#83]
[#84]
[#85]
[#86]
[#87]
[#88]
[#89]
[#90]
[#91]
[#92]
[#6]1[#6][#6]1
[#6,#7]1[#6,#7][#7]1
[!#1;*]1[!#1;*][!#1;!#6;*]1
[#6]1[#6][#6][#6]1
[#6,#7]1[#6,#7][#6,#7][#7]1
[!#1;*]1[!#1;*][!#1;*][!#1;!#6;*]1
[#6]1[#6][#6][#6][#6]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#7]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[#6]1[#6][#6][#6][#6][#6]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[#6]1[#6][#6][#6][#6][#6][#6]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[#6]1[#6][#6][#6][#6][#6][#6][#6]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[#6]1[#6][#6][#6][#6][#6][#6][#6][#6]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[#6]1[#6][#6][#6][#6][#6][#6][#6][#6][#6]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[R;a]
[R;a&!c]
[Li]-[#1]
[Li]-[Li]
[Li]-B
[Li]-C
[Li]-O
[Li]-F
[Li]-P
[Li]-S
[Li]-Cl
B-[#1]
B-B
B-C
B-N
B-O
```

```
B-F
B-[Si]
B-P
B-S
B-Cl
B-Br
C-[#1]
C-C
C-N
C-O
C-F
C-[Na]
C-[Mg]
C-[Al]
C-[Si]
C-P
C-S
C-Cl
C-[As]
C-[Se]
C-Br
C-I
N-[#1]
N-N
N-O
N-F
N-[Si]
N-P
N-S
N-Cl
N-Br
O-[#1]
O-O
O-[Mg]
O-[Na]
O-[Al]
O-[Si]
O-P
O-[K]
F-P
F-S
[Al]-[#1]
[Al]-Cl
[Si]-[#1]
[Si]-[Si]
[Si]-Cl
P-[#1]
P-P
[As]-[#1]
[As]-[As]
C(~Br)(~C)
C(~Br)(~C)(~C)
C(~Br)(~[#1])
C(~Br)(:C)
C(~Br)(:N)
C(~C)(~C)
C(~C)(~C)(~C)
C(~C)(~C)(~C)(~C)
C(~C)(~C)(~C)(~[#1])
C(~C)(~C)(~C)(~N)
C(~C)(~C)(~C)(~O)
C(~C)(~C)(~[#1])(~N)
C(~C)(~C)(~[#1])(~O)
C(~C)(~C)(~N)
C(~C)(~C)(~O)
C(~C)(~Cl)
C(~C)(~Cl)(~[#1])
C(~C)(~[#1])
C(~C)(~[#1])(~N)
C(~C)(~[#1])(~O)
C(~C)(~[#1])(~O)(~O)
C(~C)(~[#1])(~P)
```

```
C(~C)(~[#1])(~S)
C(~C)(~I)
C(~C)(~N)
C(~C)(~O)
C(~C)(~S)
C(~C)(~[Si])
C(~C)(:C)
C(~C)(:C)(:C)
C(~C)(:C)(:N)
C(~C)(:N)
C(~C)(:N)(:N)
C(~Cl)(~Cl)
C(~Cl)(~[#1])
C(~Cl)(:C)
C(~F)(~F)
C(~F)(:C)
C(~[#1])(~N)
C(~[#1])(~O)
C(~[#1])(~O)(~O)
C(~[#1])(~S)
C(~[#1])(~[Si])
C(~[#1])(:C)
C(~[#1])(:C)(:C)
C(~[#1])(:C)(:N)
C(~[#1])(:N)
C(~[#1])(~[#1])(~[#1])
C(~N)(~N)
C(~N)(:C)
C(~N)(:C)(:C)
C(~N)(:C)(:N)
C(~N)(:N)
C(~O)(~O)
C(~O)(:C)
C(~O)(:C)(:C)
C(~S)(:C)
C(:C)(:C)
C(:C)(:C)(:C)
C(:C)(:C)(:N)
C(:C)(:N)
C(:C)(:N)(:N)
C(:N)(:N)
N(~C)(~C)
N(~C)(~C)(~C)
N(~C)(~C)(~[#1])
N(~C)(~[#1])
N(~C)(~[#1])(~N)
N(~C)(~O)
N(~C)(:C)
N(~C)(:C)(:C)
N(~[#1])(~N)
N(~[#1])(:C)
N(~[#1])(:C)(:C)
N(~O)(~O)
N(~O)(:O)
N(:C)(:C)
N(:C)(:C)(:C)
O(~C)(~C)
O(~C)(~[#1])
O(~C)(~P)
O(~[#1])(~S)
O(:C)(:C)
P(~C)(~C)
P(~O)(~O)
S(~C)(~C)
S(~C)(~[#1])
S(~C)(~O)
[Si](~C)(~C)
C=C
C#C
C=N
C#N
C=O
```

C=S
N=N
N=O
N=P
P=O
P=P
C(#C)(-C)
C(#C)(-[#1])
C(#N)(-C)
C(-C)(-C)(=C)
C(-C)(-C)(=N)
C(-C)(-C)(=O)
C(-C)(-Cl)(=O)
C(-C)(-[#1])(=C)
C(-C)(-[#1])(=N)
C(-C)(-[#1])(=O)
C(-C)(-N)(=C)
C(-C)(-N)(=N)
C(-C)(-N)(=O)
C(-C)(-O)(=O)
C(-C)(=C)
C(-C)(=N)
C(-C)(=O)
C(-Cl)(=O)
C(-[#1])(-N)(=C)
C(-[#1])(=C)
C(-[#1])(=N)
C(-[#1])(=O)
C(-N)(=C)
C(-N)(=N)
C(-N)(=O)
C(-O)(=O)
N(-C)(=C)
N(-C)(=O)
N(-O)(=O)
P(-O)(=O)
S(-C)(=O)
S(-O)(=O)
S(=O)(=O)
C-C-C#C
O-C-C=N
O-C-C=O
N:C-S-[#1]
N-C-C=C
O=S-C-C
N#C-C=C
C=N-N-C
O=S-C-N
S-S-C:C
C:C-C=C
S:C:C:C
C:N:C-C
S-C:N:C
S:C:C:N
S-C=N-C
C-O-C=C
N-N-C:C
S-C=N-[#1]
S-C-S-C
C:S:C-C
O-S-C:C
C:N-C:C
N-S-C:C
N-C:N:C
N:C:C:N
N-C:N:N
N-C=N-C
N-C=N-[#1]
N-C-S-C
C-C-C=C
C-N:C-[#1]
N-C:O:C

O=C-C:C
O=C-C:N
C-N-C:C
N:N-C-[#1]
O-C:C:N
O-C=C-C
N-C:C:N
C-S-C:C
Cl-C:C-C
N-C=C-[#1]
Cl-C:C-[#1]
N:C:N-C
Cl-C:C-O
C-C:N:C
C-C-S-C
S=C-N-C
Br-C:C-C
[#1]-N-N-[#1]
S=C-N-[#1]
C-[As]-O-[#1]
S:C:C-[#1]
O-N-C-C
N-N-C-C
[#1]-C=C-[#1]
N-N-C-N
O=C-N-N
N=C-N-C
C=C-C:C
C:N-C-[#1]
C-N-N-[#1]
N:C:C-C
C-C=C-C
[As]-C:C-[#1]
Cl-C:C-Cl
C:C:N-[#1]
[#1]-N-C-[#1]
Cl-C-C-Cl
N:C-C:C
S-C:C-C
S-C:C-[#1]
S-C:C-N
S-C:C-O
O=C-C-C
O=C-C-N
O=C-C-O
N=C-C-C
N=C-C-[#1]
C-N-C-[#1]
O-C:C-C
O-C:C-[#1]
O-C:C-N
O-C:C-O
N-C:C-C
N-C:C-[#1]
N-C:C-N
O-C-C:C
N-C-C:C
Cl-C-C-C
Cl-C-C-O
C:C-C:C
O=C-C=C
Br-C-C-C
N=C-C=C
C=C-C-C
N:C-O-[#1]
O=N-C:C
O-C-N-[#1]
N-C-N-C
Cl-C-C=O
Br-C-C=O
O-C-O-C
C=C-C=C

C:C-O-C
O-C-C-N
O-C-C-O
N#C-C-C
N-C-C-N
C:C-C-C
[#1]-C-O-[#1]
N:C:N:C
O-C-C=C
O-C-C:C-C
O-C-C:C-O
N=C-C:C-[#1]
C:C-N-C:C
C-C:C-C:C
O=C-C-C-C
O=C-C-C-N
O=C-C-C-O
C-C-C-C-C
Cl-C:C-O-C
C:C-C=C-C
C-C:C-N-C
C-S-C-C-C
N-C:C-O-[#1]
O=C-C-C=O
C-C:C-O-C
C-C:C-O-[#1]
Cl-C-C-C-C
N-C-C-C-C
N-C-C-C-N
C-O-C-C=C
C:C-C-C-C
N=C-N-C-C
O=C-C-C:C
Cl-C:C:C-C
[#1]-C-C=C-[#1]
N-C:C:C-C
N-C:C:C-N
O=C-C-N-C
C-C:C:C-C
C-O-C-C:C
O=C-C-O-C
O-C:C-C-C
N-C-C-C:C
C-C-C-C:C
Cl-C-C-N-C
C-O-C-O-C
N-C-C-N-C
N-C-O-C-C
C-N-C-C-C
C-C-O-C-C
N-C-C-O-C
C:C:N:N:C
C-C-C-O-[#1]
C:C-C-C:C
O-C-C=C-C
C:C-O-C-C
N-C:C:C:N
O=C-O-C:C
O=C-C:C-C
O=C-C:C-N
O=C-C:C-O
C-O-C:C-C
O=[As]-C:C:C
C-N-C-C:C
S-C:C:C-N
O-C:C-O-C
O-C:C-O-[#1]
C-C-O-C:C
N-C-C:C-C
C-C-C:C-C
N-N-C-N-[#1]
C-N-C-N-C

O-C-C-C-C
O-C-C-C-N
O-C-C-C-O
C=C-C-C-C
O-C-C-C=C
O-C-C-C=O
[#1]-C-C-N-[#1]
C-C=N-N-C
O=C-N-C-C
O=C-N-C-[#1]
O=C-N-C-N
O=N-C:C-N
O=N-C:C-O
O=C-N-C=O
O-C:C:C-C
O-C:C:C-N
O-C:C:C-O
N-C-N-C-C
O-C-C-C:C
C-C-N-C-C
C-N-C:C-C
C-C-S-C-C
O-C-C-N-C
C-C=C-C-C
O-C-O-C-C
O-C-C-O-C
O-C-C-O-[#1]
C-C=C-C=C
N-C:C-C-C
C=C-C-O-C
C=C-C-O-[#1]
C-C:C-C-C
Cl-C:C-C=O
Br-C:C:C-C
O=C-C=C-C
O=C-C=C-[#1]
O=C-C=C-N
N-C-N-C:C
Br-C-C-C:C
N#C-C-C-C
C-C=C-C:C
C-C-C=C-C
C-C-C-C-C-C
O-C-C-C-C-C
O-C-C-C-C-O
O-C-C-C-C-N
N-C-C-C-C-C
O=C-C-C-C-C
O=C-C-C-C-N
O=C-C-C-C-O
O=C-C-C-C=O
C-C-C-C-C-C-C
O-C-C-C-C-C-C
O-C-C-C-C-C-O
O-C-C-C-C-C-N
O=C-C-C-C-C-C
O=C-C-C-C-C-O
O=C-C-C-C-C=O
O=C-C-C-C-C-N
C-C-C-C-C-C-C-C
C-C-C-C-C-C(C)-C
O-C-C-C-C-C-C-C
O-C-C-C-C-C(C)-C
O-C-C-C-C-C-O-C
O-C-C-C-C-C(O)-C
O-C-C-C-C-C-N-C
O-C-C-C-C-C(N)-C
O=C-C-C-C-C-C-C
O=C-C-C-C-C(O)-C
O=C-C-C-C-C(=O)-C
O=C-C-C-C-C(N)-C
C-C(C)-C-C

C-C(C)-C-C-C
C-C-C(C)-C-C
C-C(C)(C)-C-C
C-C(C)-C(C)-C
Cc1ccc(C)cc1
Cc1ccc(O)cc1
Cc1ccc(S)cc1
Cc1ccc(N)cc1
Cc1ccc(Cl)cc1
Cc1ccc(Br)cc1
Oc1ccc(O)cc1
Oc1ccc(S)cc1
Oc1ccc(N)cc1
Oc1ccc(Cl)cc1
Oc1ccc(Br)cc1
Sc1ccc(S)cc1
Sc1ccc(N)cc1
Sc1ccc(Cl)cc1
Sc1ccc(Br)cc1
Nc1ccc(N)cc1
Nc1ccc(Cl)cc1
Nc1ccc(Br)cc1
Clc1ccc(Cl)cc1
Clc1ccc(Br)cc1
Brc1ccc(Br)cc1
Cc1cc(C)ccc1
Cc1cc(O)ccc1
Cc1cc(S)ccc1
Cc1cc(N)ccc1
Cc1cc(Cl)ccc1
Cc1cc(Br)ccc1
Oc1cc(O)ccc1
Oc1cc(S)ccc1
Oc1cc(N)ccc1
Oc1cc(Cl)ccc1
Oc1cc(Br)ccc1
Sc1cc(S)ccc1
Sc1cc(N)ccc1
Sc1cc(Cl)ccc1
Sc1cc(Br)ccc1
Nc1cc(N)ccc1
Nc1cc(Cl)ccc1
Nc1cc(Br)ccc1
Clc1cc(Cl)ccc1
Clc1cc(Br)ccc1
Brc1cc(Br)ccc1
Cc1c(C)cccc1
Cc1c(O)cccc1
Cc1c(S)cccc1
Cc1c(N)cccc1
Cc1c(Cl)cccc1
Cc1c(Br)cccc1
Oc1c(O)cccc1
Oc1c(S)cccc1
Oc1c(N)cccc1
Oc1c(Cl)cccc1
Oc1c(Br)cccc1
Sc1c(S)cccc1
Sc1c(N)cccc1
Sc1c(Cl)cccc1
Sc1c(Br)cccc1
Nc1c(N)cccc1
Nc1c(Cl)cccc1
Nc1c(Br)cccc1
Clc1c(Cl)cccc1
Clc1c(Br)cccc1
Brc1c(Br)cccc1
CC1CCC(C)CC1
CC1CCC(O)CC1
CC1CCC(S)CC1
CC1CCC(N)CC1
CC1CCC(Cl)CC1

CC1CCC(Br)CC1
OC1CCC(O)CC1
OC1CCC(S)CC1
OC1CCC(N)CC1
OC1CCC(Cl)CC1
OC1CCC(Br)CC1
SC1CCC(S)CC1
SC1CCC(N)CC1
SC1CCC(Cl)CC1
SC1CCC(Br)CC1
NC1CCC(N)CC1
NC1CCC(Cl)CC1
NC1CCC(Br)CC1
ClC1CCC(Cl)CC1
ClC1CCC(Br)CC1
BrC1CCC(Br)CC1
CC1CC(C)CCC1
CC1CC(O)CCC1
CC1CC(S)CCC1
CC1CC(N)CCC1
CC1CC(Cl)CCC1
CC1CC(Br)CCC1
OC1CC(O)CCC1
OC1CC(S)CCC1
OC1CC(N)CCC1
OC1CC(Cl)CCC1
OC1CC(Br)CCC1
SC1CC(S)CCC1
SC1CC(N)CCC1
SC1CC(Cl)CCC1
SC1CC(Br)CCC1
NC1CC(N)CCC1
NC1CC(Cl)CCC1
NC1CC(Br)CCC1
ClC1CC(Cl)CCC1
ClC1CC(Br)CCC1
BrC1CC(Br)CCC1
CC1C(C)CCCC1
CC1C(O)CCCC1
CC1C(S)CCCC1
CC1C(N)CCCC1
CC1C(Cl)CCCC1
CC1C(Br)CCCC1
OC1C(O)CCCC1
OC1C(S)CCCC1
OC1C(N)CCCC1
OC1C(Cl)CCCC1
OC1C(Br)CCCC1
SC1C(S)CCCC1
SC1C(N)CCCC1
SC1C(Cl)CCCC1
SC1C(Br)CCCC1
NC1C(N)CCCC1
NC1C(Cl)CCCC1
NC1C(Br)CCCC1
ClC1C(Cl)CCCC1
ClC1C(Br)CCCC1
BrC1C(Br)CCCC1
CC1CC(C)CC1
CC1CC(O)CC1
CC1CC(S)CC1
CC1CC(N)CC1
CC1CC(Cl)CC1
CC1CC(Br)CC1
OC1CC(O)CC1
OC1CC(S)CC1
OC1CC(N)CC1
OC1CC(Cl)CC1
OC1CC(Br)CC1
SC1CC(S)CC1
SC1CC(N)CC1
SC1CC(Cl)CC1

SC1CC(Br)CC1
NC1CC(N)CC1
NC1CC(Cl)CC1
NC1CC(Br)CC1
ClC1CC(Cl)CC1
ClC1CC(Br)CC1
BrC1CC(Br)CC1
CC1C(C)CCC1
CC1C(O)CCC1
CC1C(S)CCC1
CC1C(N)CCC1
CC1C(Cl)CCC1
CC1C(Br)CCC1
OC1C(O)CCC1
OC1C(S)CCC1
OC1C(N)CCC1
OC1C(Cl)CCC1
OC1C(Br)CCC1
SC1C(S)CCC1
SC1C(N)CCC1
SC1C(Cl)CCC1
SC1C(Br)CCC1
NC1C(N)CCC1
NC1C(Cl)CC1
NC1C(Br)CCC1
ClC1C(Cl)CCC1
ClC1C(Br)CCC1
BrC1C(Br)CCC1
c2ccc1ccccc1c2
C2CC1CCCC1C2
C2CCC1CCCCC1C2
C2CCC1CCCCCC1CC2
C2CCCC1CCCCCCC1CC2

K Die verwendeten reduzierten Wörterbücher

L_64_MAYSC

```
#Comments after SMARTS
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[#16]
[#17]
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#7]1
[#6]1[#6][#6][#6][#6][#6]1
[#6]1[#6][#6][#6][#6]1
[#7]
[#8]
[#9]
[R;a&!c]
[R;a]
C-C
C-C-C-C-C
C-C-C-C-C-C
C-C-C-C-C-C-C
C-C-C-C-C-C-C-C
C-C-C(C)-C-C
C-C-N-C-C
C-C(C)-C-C
C-C(C)-C-C-C
C-C(C)(C)-C-C
C-F
C-N
C-N-C-C-C
C-O
C-S
C(-C)(-N)(=O)
C(-C)(-O)(=O)
C(-C)(=C)
C(-C)(=O)
C(-N)(=O)
C(-O)(=O)
C(-C)(-C)
C(-C)(-C)(-C)
C(-C)(-C)(-C)(-C)
C(-C)(-C)(-N)
C(-C)(-C)(-O)
C(-C)(-N)
C(-C)(-O)
C(-C)(-S)
C(-F)(-F)
C(-N)(-N)
C(-O)(-O)
C=C
C=N
C=O
Cc1ccc(N)cc1
N-C-C-C-C
N-C-C-C-C-C
N-N
N(-C)(-C)
N(-C)(-C)(-C)
O-C-C-C-C
```

```
O-C-C-C-C-C
O(-C)(-C)
O=C-C-C
O=C-C-C-C
O=C-C-C-C-C
O=C-C=C
O=C-N-C-C
S(=O)(=O)
```

G_64_MAYSC

```
#Comments after SMARTS
N-C-C-N-C
[#7]
C-Cl
C(-C)(-C)(=O)
C(-O)(-O)
S=C-N-C
[#9]
N(-C)(-C)(-C)
CC1CC(O)CC1
C(-F)(-F)
Cc1ccc(C)cc1
Cc1cc(N)ccc1
CC1C(C)CCC1
O-C-C-C-C-C-C
C-C-C-C-C
O=C-C-C
Nc1c(N)cccc1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
C(-C)(-C)(-C)
Oc1ccc(N)cc1
C-C-C-C-C-C(C)-C
[#6]1[#6][#6][#6][#6]1
C(-C)(-C)(-C)(-O)
O-C-C-N-C
Cc1cc(C)ccc1
C-C
C(-C)(-C)(-N)
O-C-C-C-C-C-C-C
N-C-C-C-C
Oc1cc(N)ccc1
Oc1cc(S)ccc1
N(-C)(-C)
CC1CC(N)CCC1
C(-C)(=O)
Nc1ccc(Br)cc1
S-C=N-C
[#35]
N-C-C-N
O-C-C-C-C-C(N)-C
Cc1ccc(N)cc1
O-C-O-C
S(-C)(-O)
S(-C)(-C)
O=C-C-C-O
C-O
C(-C)(-O)
O=C-C-C-C-C-N
Cl-C-C-O
Nc1c(Br)cccc1
O-C-C-C-C-N
[#17]
C(-N)(-N)
O=C-C-C-C-C
[#6]1[#6][#6][#6][#6][#6]1
C-C-C-C-C-C
N-C=N-C
```

```
Cc1ccc(O)cc1
C-N-C-N-C
O-C-C-C-N
O-C-C=C
S(-C)(=O)
S-C-S-C
[#6,#7]1[#6,#7][#6,#7][#6,#7][#7]1
C-Cl
```

L_64_ASINEX_PC_2008

```
#Comments after SMARTS
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[#16]
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#7]1
[#6]1[#6][#6][#6][#6][#6]1
[#6]1[#6][#6][#6][#6]1
[#7]
[#8]
[#9]
[R;a&!c]
C-C
C-C-C-C-C
C-C-C-C-C-C
C-C-N-C-C
C-C-O-C-C
C-C(C)-C-C
C-N
C-N-C-C-C
C-N-C-N-C
C-O
C-S
C(-C)(-N)(=O)
C(-C)(=O)
C(-N)(=O)
C(~C)(-C)
C(~C)(-C)(~C)
C(~C)(-C)(-N)
C(~C)(-N)
C(~C)(-O)
C(~C)(-S)
C(~N)(-N)
C(~O)(-O)
C=O
C=S
Cc1c(N)cccc1
Cc1cc(C)ccc1
Cc1ccc(C)cc1
Cc1ccc(N)cc1
Cc1ccc(O)cc1
N-C-C-C-C
N-C-C-C-C-C
N-C-C-C-N
N-C-C-N
N-C-C-N-C
N-C-C-O-C
N-C-N-C
N-C-N-C-C
N(~C)(-C)
N(~C)(-C)(~C)
O-C-C-N
O-C-C-N-C
```

```
O(-C)(~C)
O=C-C-C
O=C-C-C-C
O=C-C-N
O=C-C-N-C
O=C-N-C-C
Oc1c(O)cccc1
S=C-N-C
```

G_64_ASINEX_PC_2008

```
#Comments after SMARTS
c2ccc1ccccc1c2
OC1C(N)CCCC1
Nc1cc(N)ccc1
N-C-C=C
Sc1ccc(Cl)cc1
C(~C)(~N)
Cc1cc(O)ccc1
C(~C)(~C)(~O)
C-C=N-N-C
Sc1ccc(S)cc1
Oc1cc(O)ccc1
N=C-N-C-C
O-C-C-C-C-N
Oc1c(S)cccc1
Oc1cc(N)ccc1
C-C=C-C
[#6]1[#6][#6][#6][#6][#6][#6]1
C(~C)(~C)
C(~C)(~S)
C-C(C)(C)-C-C
O-C=C-C
N-C-C-C-C
Cc1ccc(Br)cc1
Sc1cc(Cl)ccc1
Cc1ccc(C)cc1
C(~C)(~C)(~C)
O-C-C-C-C-C-N-C
CC1C(C)CCCC1
C(~C)(=N)
C-C
N-O
C-N-C-N-C
[#16]
Cc1ccc(O)cc1
O=C-C-C-C-N
O-C-C-C-C-C
N#C-C-C
CC1CC(N)CC1
C-C-S-C
[#6]1[#6][#6][#6]1
CC1CC(C)CCC1
C(~C)(~C)(~C)(~N)
[R;a]
C-N
Cc1ccc(N)cc1
CC1CCC(N)CC1
O=C-C-C-C-O
CC1C(C)CCC1
Cc1c(C)cccc1
O=S-C-C
[#9]
O-C-C-C=O
O-C-C-N-C
Nc1ccc(Cl)cc1
C(#C)(~C)
[R;a&!c]
```

```
Cc1cc(C)ccc1
C-C-C(C)-C-C
O-C-C-C-C-C-C-C
Cl-C-C=O
O-C-C-C-C
O-C-C-C-C-C-O
C(~N)(~N)
N=C-C-C
```

G_S_64_MAYSC

```
#Comments after SMARTS
N-C=N-C
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#7]1
O=C-N-N
Sc1ccc(Cl)cc1
C(~C)(~C)(~C)
[#6]1[#6][#6][#6][#6][#6][#6][#6][#6]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[#6]1[#6][#6][#6][#6][#6][#6][#6][#6][#6]1
C-C-C-C-C-C-C-C
CC1CCC(C)CC1
N(~C)(~C)
C(~C)(=C)
C-C-C-C-C
Cc1cc(C)ccc1
C(~C)(~C)(~N)
C-C=C-C-C
O-C-C=N
[#8]
O=S-C-C
C-C-C-C-C-C-C
O-C-C-C-C-C(C)-C
O=C-C-C-N
Cc1cc(Cl)ccc1
O-C-C=O
[#9]
c2ccc1ccccc1c2
N-C-C=C
N=C-N-C-C
[#6]1[#6][#6][#6][#6][#6]1
CC1CC(C)CCC1
N-C-N-C
Cc1ccc(O)cc1
C-N
C-F
O-C-C-C-C-C(O)-C
C(~N)(=N)
CC1CC(O)CC1
C-N-C-C-C
[#17]
C(~C)(~O)(=O)
O=C-N-C=O
C(~N)(~N)
N(~C)(~O)
N(~C)(=C)
CC1CC(C)CC1
O=C-C-C
Cc1ccc(C)cc1
C-O
S(~C)(~O)
C-C
C=O
O-C-C-C-C-C-N
CC1CCC(O)CC1
Cc1c(C)cccc1
Cc1ccc(N)cc1
[#6]1[#6][#6][#6][#6]1
```

```
Oc1cc(Cl)ccc1
C-C(C)-C-C-C
Brc1cc(Br)ccc1
C(-C)(-C)(=O)
Oc1cc(N)ccc1
O=C-C-C-C-C(=O)-C
C-C(C)-C-C
CC1CC(O)CC1
```

G_S_64_ASINEX_PC_2008

```
#Comments after SMARTS
[#6]1[#6][#6][#6][#6][#6]1
O-C-C-C-C-C
C-C-C-C-C
Cc1ccc(Br)cc1
N(-C)(-C)(-C)
O=C-N-C-C
S=C-N-C
N-C-N-C
C=C
C(-C)(-C)
N-C-C-C-C
O-C-C-C-C-C-N
C(-F)(-F)
Cc1c(S)cccc1
N-C-C-C-C-C
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
C(#N)(-C)
Cc1c(C)cccc1
Cc1cc(C)ccc1
C-S
C(-C)(-N)
C=N-N-C
N-C-S-C
N-C-S-C
C-C-C=C
[#7]
Sc1ccc(Cl)cc1
C-C
Nc1cc(N)ccc1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#7]1
Cc1cc(N)ccc1
Oc1c(N)cccc1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[#35]
Nc1ccc(Cl)cc1
Cc1ccc(N)cc1
O=C-C-C-C-C
N-C-C=C
C(-C)(-S)
CC1CC(C)CCC1
[#9]
Cc1c(O)cccc1
N-C-N-C-C
O=C-C-C-C=O
C-N-C-C-C
Cc1ccc(O)cc1
O(-C)(-C)
O=C-N-C-C
Oc1cc(O)ccc1
C-O
C-N
C-C=C-C-C
C(-C)(-C)(-C)
C-C=N-N-C
O-C-C-C-C-N
Oc1ccc(N)cc1
```

```
N-C-C-O-C
Nc1ccc(Cl)cc1
C-C(C)-C-C
Cc1ccc(C)cc1
Cl-C-C=O
[#16]
[R;a&!c]
C(-C)(-N)(=C)
```

L_S_64_MAYSC

```
#Comments after SMARTS
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[#16]
[#17]
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#7]1
[#6]1[#6][#6][#6][#6][#6][#6][#6][#6][#6]1
[#6]1[#6][#6][#6][#6][#6]1
[#6]1[#6][#6][#6][#6]1
[#7]
[#8]
[#9]
[R;a&!c]
[R;a]
C-C
C-C-C-C-C
C-C-C-C-C-C
C-C-C-C-C-C-C
C-C-C-C-C-C-C-C
C-C-C-C-C-C(C)-C
C-C-C(C)-C-C
C-C-N-C-C
C-C(C)-C-C
C-C(C)-C-C-C
C-C(C)(C)-C-C
C-N
C-N-C-C-C
C-O
C-S
C(-C)(-N)(=O)
C(-C)(-O)(=O)
C(-C)(=C)
C(-C)(=O)
C(-N)(=O)
C(-O)(=O)
C(-C)(-C)
C(-C)(-C)(-C)
C(-C)(-C)(-C)(-C)
C(-C)(-C)(-N)
C(-C)(-C)(-O)
C(-C)(-N)
C(-C)(-O)
C(-C)(-S)
C(-N)(-N)
C(-O)(-O)
C=C
C=N
C=O
CC1CC(C)CCC1
N-C-C-C-C
N-C-C-C-C-C
N-N
N(-C)(-C)
O-C-C-C-C
O-C-C-C-C-C
O-C-C-C-C-C-C
```

```
O(-C)(-C)
O=C-C-C
O=C-C-C-C
O=C-C-C-C-C
O=C-C-C-C-C-C
O=C-C-C-C-C-C-C
O=C-C=C
O=C-N-C-C
```

L_S_64_ASINEX_PC_2008

```
#Comments after SMARTS
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[#16]
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#7]1
[#6]1[#6][#6][#6][#6][#6]1
[#6]1[#6][#6][#6][#6]1
[#7]
[#8]
[R;a&!c]
[R;a]
C-C
C-C-C-C-C
C-C-C-C-C-C
C-C-N-C-C
C-C-O-C-C
C-C(C)-C-C
C-N
C-N-C-C-C
C-N-C-N-C
C-O
C-S
C(-C)(-N)(=O)
C(-C)(=O)
C(-N)(=O)
C(-C)(-C)
C(-C)(-C)(-C)
C(-C)(-C)(-N)
C(-C)(-C)(-O)
C(-C)(-N)
C(-C)(-O)
C(-C)(-S)
C(-N)(-N)
C(-O)(-O)
C=O
C=S
Cc1cc(C)ccc1
Cc1cc(O)ccc1
Cc1ccc(C)cc1
Cc1ccc(O)cc1
N-C-C-C-C
N-C-C-C-C-C
N-C-C-C-N
N-C-C-N
N-C-C-N-C
N-C-C-O-C
N-C-N-C
N-C-N-C-C
N(-C)(-C)
N(-C)(-C)(-C)
O-C-C-N
O-C-C-N-C
```

```
O(~C)(-C)
O=C-C-C
O=C-C-C-C
O=C-C-N
O=C-C-N-C
O=C-N-C-C
Oc1c(O)cccc1
S=C-N-C
```

L_S_64_BBS

```
#Comments after SMARTS
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[!#1;*]1[!#1;*][!#1;*][!#1;*][!#1;*][!#1;!#6;*]1
[#16]
[#17]
[#6,#7]1[#6,#7][#6,#7][#6,#7][#6,#7][#7]1
[#6,#7]1[#6,#7][#6,#7][#6,#7][#7]1
[#6]1[#6][#6][#6][#6][#6]1
[#7]
[#8]
[#9]
[R;a&!c]
[R;a]
C-C
C-C-C-C-C
C-C-C-C-C-C
C-C-C-C-C-C-C
C-C-C-C-C-C-C-C
C-C-C(C)-C-C
C-C-N-C-C
C-C-O-C-C
C-C(C)-C-C
C-C(C)-C-C-C
C-N
C-N-C-C-C
C-O
C#N
C(-C)(-N)(=O)
C(-C)(-O)(=O)
C(-C)(=O)
C(-N)(=O)
C(-O)(=O)
C(#N)(-C)
C(-C)(-C)
C(-C)(-C)(-C)
C(-C)(-C)(-N)
C(-C)(-C)(-O)
C(-C)(-N)
C(-C)(-O)
C(-O)(-O)
C=O
Cc1c(O)cccc1
Cc1ccc(C)cc1
Cc1ccc(O)cc1
N-C-C-C-C
N-C-C-C-C-C
N-C-C-C-N
N-C-C-N
N-C-C-N-C
N(-C)(-C)
N(-C)(-C)(-C)
O-C-C-C-C
O-C-C-C-C-C
O-C-C-C-C-C-C
O-C-C-C-C-C-C-C
O-C-C-N
O-C-C-N-C
```

```
O(-C)(~C)
O=C-C-C
O=C-C-C-C
O=C-C-C-C-C
O=C-C-C-C-C-C
O=C-C-C-C-C-C-C
O=C-C-N
O=C-N-C-C
```